Jazz Up Your Web Site with JavaServer Pages

REVIEWS
- Tango 2000
- JFC Suite 2.1

JAVAPro™

www.java-pro.com
A Fawcette Technical Publication
February 2000, Vol. 4, No. 2

Build an Icon Editor

Give NT a Java Jolt

Harness the Power of MTS for Java

Bridge Java & COM with XML

Fast Data Access with JDBC 2.0

Free Source Code

Code Online
www.java-pro.com

U.S. $5.95 (Canada $7.95)

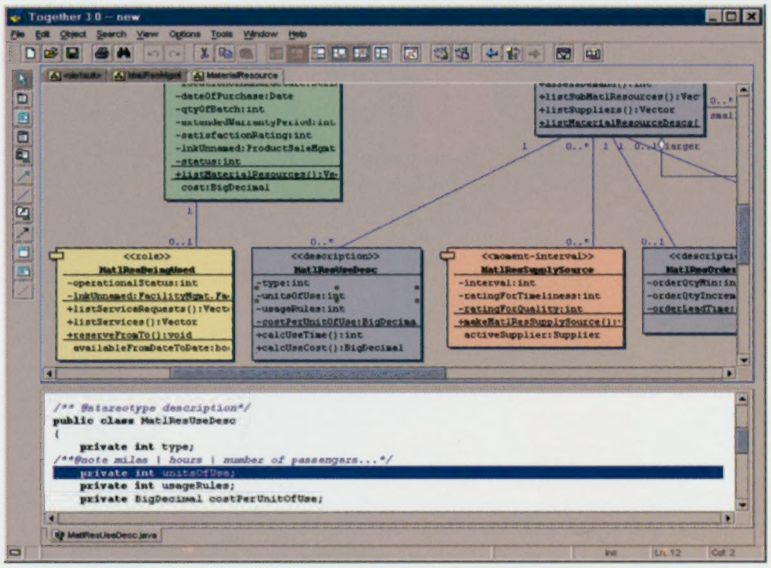
02>

7 25274 91607 8

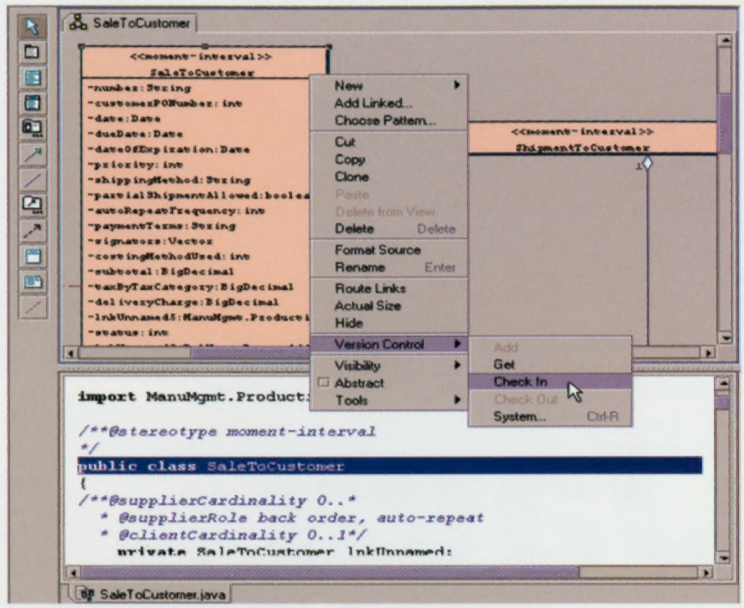
New!

Together/J's EJBs, GoF patterns, and simultaneous round-trip engineering—what Home Depot calls "the backbone of all our Java development"

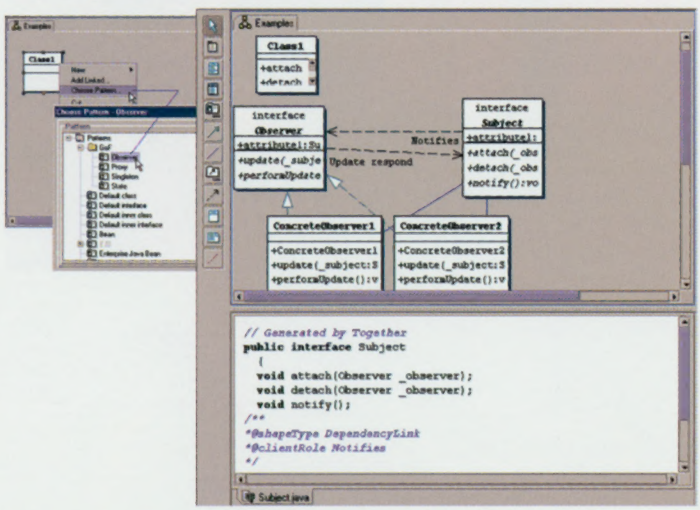
Download the free TJ Whiteboard and experience TJ for yourself



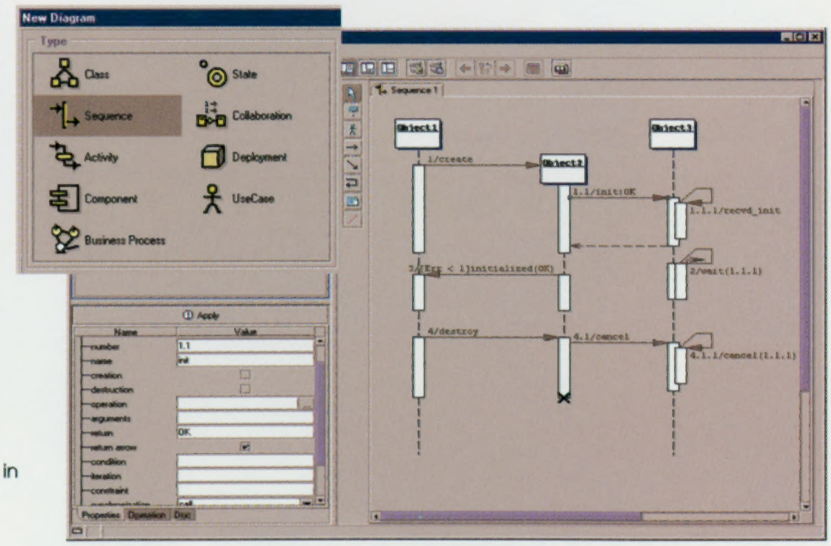
OneSource(tm) simultaneous round-trip engineering—always in-sync always up-to-date, complete freedom to work visually or textually as you see fit



BigPlay(tm) version-control features are a just mouse-click away—complete end-to-end versioning of models, source, and documentation across the enterprise.



Select a class, choose EJB™ or a GoF pattern, and TJ auto-generates it for you – in Java or C++ (or both, in Together/Enterprise)



UML 1.3 (nine kinds of diagrams) including reverse engineering of methods into sequence diagrams. Plus HTML and RaftMaker(tm) RTF documentation—invoke it as part of your daily build, deliver always-up-to-date doc to your boss

Download TJ Whiteboard from www.togethersoft.com/jp/

"Having seen Together/J's direct use of Java code to support the models, I am forced to wonder why anyone would like to work in any other way! I do think that the strength of the product lies in its interoperability — it's a 'complete solution', not just a modeling tool." — Jon Collins, Senior Analyst, **Bloor Research**

"I really like Together/J. No other vender has even come close to making such an intuitive and convenient diagramming interface." —David Marine, Sr. Web Applications Developer, **NetSolve**

"We discovered in our evaluation that other modeling tools only get used during the front-end of a project. Yet we found Together/J to be the one UML modeler that our developers will use from start to finish. Why? It's the only one that delivers simultaneous (rather than batch-mode) round-trip engineering." — Curtis Chambers, **The Home Depot**

"I'm sold... I just finished doing the entire Use Case diagram for the project, (in the format I wanted it), publishing it to a web document on our web site, am working on the sequence and collaboration diagrams right now; and am scheduling a net video discussion group of all our users to discuss the use case designs that we posted... All with Together/J. Your product does everything we need to do now and more..." — Steve McConnell, **The Shaw Group**

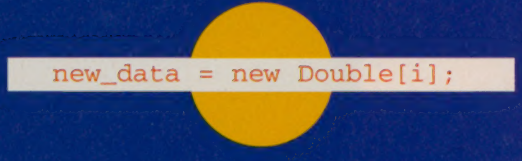
"Together/J is an excellent modeling CASE tool for Java developers... It also meets the needs of real-world object modelers. Its support for language-specific features such as inner classes and JavaBeans, puts it miles ahead of other CASE tools that merely support the Java-language syntax... Together/J 3's support for key UML 1.3 diagrams, patterns, and Java-specific language features makes it a valuable addition to your Java development toolset." — **Jolt Award '99**

TogetherSoft LLC
1720 Leigh Drive
Raleigh, NC 27603 USA
(919) 772-9350
www.togethersoft.com

©1999 TogetherSoft LLC. All rights reserved.
Together is a registered trademark of TogetherSoft LLC
Java and EJB are trademarks of Sun Microsystems, Inc.

Profiling with JProbe Profiler

There's a problem in this line of code



Your code may look harmless enough, but bottlenecks may be lurking in the most unexpected places. Pinpointing these problems can be hard ... especially if your profiler only looks at methods. JProbe Profiler with integrated Memory Debugger highlights the precise line of code responsible for performance problems, not just the method. So make it easier on yourself with JProbe: find the line.

JPROBE Profiler

www.klgroup.com/performance

KL GROUP
Software Development Productivity

KL Group Inc. Toll-Free: 1-888-328-9599 Tel: 416-594-1026 Fax: 416-594-1919
KL Group Europe Tel: +31 (0)20 679 95 03 Fax: +31 (0)20 470 03 26
www.klgroup.com

© Copyright 1999 by KL Group Inc. All rights reserved. KL Group and the KL Group logo are registered trademarks of KL Group Inc. JProbe, JProbe Profiler and JProbe Memory Debugger are trademarks of KL Group Inc. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in US and other countries. All other products are trademarks or registered trademarks of their respective companies.
JP001

Wish you had the power of Excel on the Web?

Copyright © 1999 Tidestone Technologies, Inc. All rights reserved. Tidestone, the Tidestone logo, and Formula One are
logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All other

Let's face it, you can wish all you want but Excel is still a heavy Windows-based, desktop-bound application—not your ideal choice for building online solutions.

Formula One, however, is an Excel-compatible spreadsheet toolset designed for distributed computing. Its robust calculating, formatting, and charting abilities, combined with point-and-click access available anywhere via a browser, make Formula One the most powerful spreadsheet technology available for integration into your Web deployed applications.

So whether you need database reporting, analysis, calculations, data presentation, or data entry, Formula One has the following advantages:

- **Formula One is built in the Java programming language and is ideal for widespread distribution on the Web.** Excel and Office Web Components are designed for use behind a firewall and require Microsoft Office to be installed on all desktops.
- **Formula One's architecture and JDBC methods enable it to be used with a wide variety of database and application servers.** Excel can't.
- **Formula One provides a JavaBean and applet with an API of more than 400 properties, methods, and events.** Excel is not an API-driven application and can not be used as a component in a Java application.
- **Formula One has a lightweight footprint.** System requirements for Excel 2000 include 146 MB of hard disk space. A Formula One JAR file is approximately 1 MB.
- **Formula One writes files optimized for the Web.** Formula One's file compression technology is built with distributed computing in mind, enabling it to save files up to 90% smaller than Excel's in some cases. Excel's overhead caused by its dependence on Office and Windows produces files not suited for Web-based computing.
- **Formula One is 100% Pure Java and only requires the proper virtual machine to function.** Excel requires Windows to operate.

FORMULA ONE 7.0

Java Spreadsheet Toolset For Building Web Applications

- *Cross-platform*
- *Light footprint*
- *Extensive API*
- *JDBC methods*
- *Excel compatible*
- *Charting*

A PRODUCT OF Tidestone TECHNOLOGIES

- Free Trials
- Live Demos
- Free Sample Code

call: 800-884-8665
or download at: www.tidestone.com

...of Tidestone Technologies, Inc. Java, 100% Pure Java, and all Java-based trademarks and ...are property of their respective owners. Specifications subject to change without notice.

500,000 downloads of Sun Microsystems' Java™ Foundation Classes.

Multiple platforms.

One installer.

InstallAnywhere®

ENTERPRISE EDITION

www.ZeroG.com

Deploying enterprise software across multiple platforms? Whether they're applications, libraries, or servlets written in Java or platform specific code, InstallAnywhere is the only solution that creates a single, universal installer that can deploy software from the Internet or CD-ROM to virtually any platform, including Windows, Solaris, Linux, AIX, and Mac OS. Try it today and find out why companies such as Sun Microsystems, Adobe and FedEx are using InstallAnywhere to ensure the successful deployment of their software across multiple platforms.

Any application. Any platform. The only solution.

zero G
ZERO G SOFTWARE

© 1999 Zero G Software, Inc. InstallAnywhere, Zero G Software, the Zero G logo, and ZeroG.com are trademarks or registered trademarks of Zero G Software, Inc. Sun, Sun Microsystems, Java, Java Foundation Classes, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All other trademarks and registered trademarks are the property of their respective holders.

JAVA Pro™

February 2000,
Vol. 4, No. 2

Contents

Features

18 Transform MTS into a Pure Java Object Monitor
by Marcus Daley
Microsoft's Transaction Server is powerful. How can you harness this power for Java? Marcus explains that it's not as difficult as you might think to enable MTS to become a Java component-hosting environment.

28 Build Faster, More Flexible Databases with JDBC 2.0
by Brett Spell
Although JDBC 2.0 is not a new specification, most drivers didn't implement it until recently. In this, the first of three articles examining the many features of JDBC 2.0, Brett examines the areas where it shows a marked improvement over the 1.2 release.

34 Java to COM+: Making Cross-Platform Accessibility Work
by Bill Block and Jerry Brunning
Getting your Java apps to talk to your COM+ apps might be messy, but it's possible. Here's how our authors reconfigured their architecture to make it happen.

Reviews

12 Tango 2000
14 JFC Suite 2.1

Columns

40 Javatecture
by James Cooper
Most response pages are downright ugly. James tells how to beautify yours using JavaServer Pages.

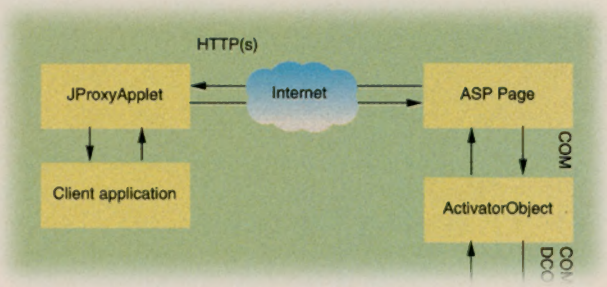
46 Visual Components
by Claude Duguay
Claude unveils a full-function icon editor that allows users to customize icons in IDEs and graphic applications.

60 Impure Thoughts
by Rick Grehan
Sparse matrices can be problematic in linear programming. Rick explores techniques that help you deal with them without exhausting your memory.

72 Ask the Java Pro
by Daniel F. Savarese
Automatic method inlining can throw a monkey wrench in Java's "write once run anywhere" credo. Daniel discusses this as well as why cross-platform testing is just as important in Java as in any other language.

Departments

9 Editor's Note
by Sean Gallagher
16 New Products
71 Ad Index



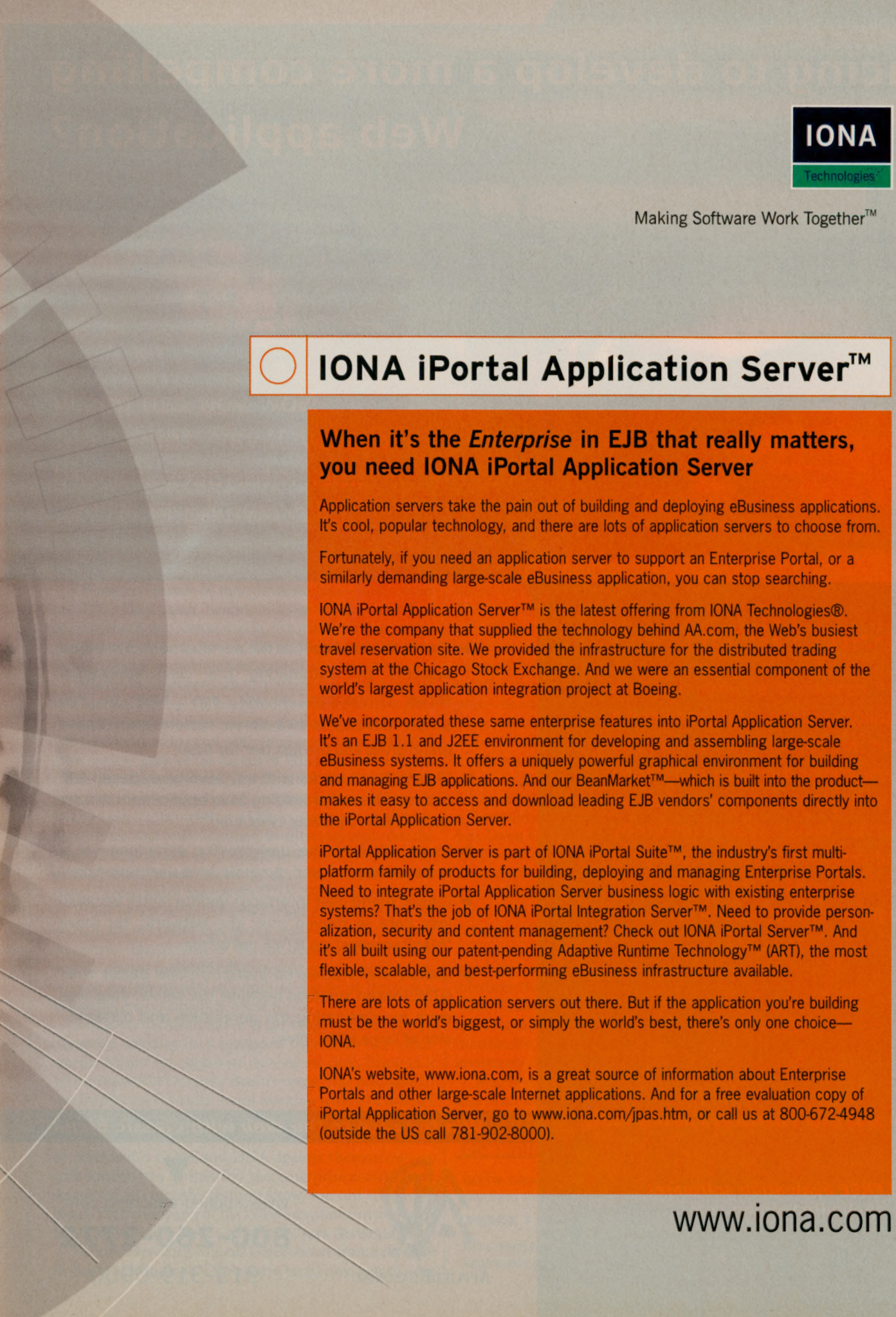
Your pure Java application can talk to a set of COM+ components; see page 34.

BPA INTERNATIONAL

PRINTED IN USA

Java Pro is published monthly for $34.97 per year, by Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA, USA, 94301-2500. Tel. 650-833-7100; Fax. 650-853-0230. **Postmaster:** Send address changes to *Java Pro*, P.O. Box 54584, Boulder, CO 80322-4584. Application to mail at Periodicals Postage Rates is pending at Palo Alto, CA and additional mailing offices. **© 2000 Fawcette Technical Publications**, all rights reserved. *Java Pro* is an independent publication not affiliated with Sun Microsystems. Sun Microsystems is not responsible in any way for the editorial policy or other contents of the publication. All contents of *Java Pro* are copyright © 2000 by Fawcette Technical Publications, unless otherwise noted. "VBITS" and "Interactive Developer" are trademarks of Fawcette Technical Publications, a California Corporation, James E. Fawcette, President. Java is a trademark of Sun Microsystems. Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion with no intention of infringement of the trademark. Although all reasonable attempts are made to ensure accuracy, the publisher does not assume any liability for errors or omissions anywhere in the publication. **Canadian GST#: 86831 0764 ISSN: 1096-4495.**

Download a free evaluation copy of iPortal Application Server at
www.iona.com/jpas.htm



Making Software Work Together™

IONA iPortal Application Server™

When it's the *Enterprise* in EJB that really matters, you need IONA iPortal Application Server

Application servers take the pain out of building and deploying eBusiness applications. It's cool, popular technology, and there are lots of application servers to choose from.

Fortunately, if you need an application server to support an Enterprise Portal, or a similarly demanding large-scale eBusiness application, you can stop searching.

IONA iPortal Application Server™ is the latest offering from IONA Technologies®. We're the company that supplied the technology behind AA.com, the Web's busiest travel reservation site. We provided the infrastructure for the distributed trading system at the Chicago Stock Exchange. And we were an essential component of the world's largest application integration project at Boeing.

We've incorporated these same enterprise features into iPortal Application Server. It's an EJB 1.1 and J2EE environment for developing and assembling large-scale eBusiness systems. It offers a uniquely powerful graphical environment for building and managing EJB applications. And our BeanMarket™—which is built into the product—makes it easy to access and download leading EJB vendors' components directly into the iPortal Application Server.

iPortal Application Server is part of IONA iPortal Suite™, the industry's first multi-platform family of products for building, deploying and managing Enterprise Portals. Need to integrate iPortal Application Server business logic with existing enterprise systems? That's the job of IONA iPortal Integration Server™. Need to provide personalization, security and content management? Check out IONA iPortal Server™. And it's all built using our patent-pending Adaptive Runtime Technology™ (ART), the most flexible, scalable, and best-performing eBusiness infrastructure available.

There are lots of application servers out there. But if the application you're building must be the world's biggest, or simply the world's best, there's only one choice—IONA.

IONA's website, www.iona.com, is a great source of information about Enterprise Portals and other large-scale Internet applications. And for a free evaluation copy of iPortal Application Server, go to www.iona.com/jpas.htm, or call us at 800-672-4948 (outside the US call 781-902-8000).

www.iona.com

Racing to develop a more compelling Web application?

Now you can easily deploy Java™ thin-client user interfaces instead of drowning in HTML.

Classic Blend™

Classic Blend™ consists of an ultra-thin Java client that displays your user interface and a communication mechanism for connecting the client to any middle-tier Java application server. Classic Blend™ is the first tool to support a true ultra-thin client architecture by its unique functional partitioning, placing the display on the client and the presentation and application logic on the application server.

- Classic Blend simplifies the development of *n*-tiered systems by automating the client-application server connection...so you don't have to. You simply develop a two-tiered system and Classic Blend gives you the last tier for free!
- Classic Blend works with any Java bean-based IDE and complements leading Java application servers such as BEA Systems WebLogic®.
- Users will love the desktop-like experience of a more interactive and dynamic Java user interface.
- Classic Blend user interfaces update faster because less data is moved over the network. No more "blank screens" as HTML refreshes.
- Classic Blend-powered systems scale better than HTML-based systems because the server does less work by eliminating the generation and transmission of large HTML files.

Classic Blend™
by APPLIEDREASONING

Making Ultra-Thin Java User Interfaces a Reality Today.

Win the race to the Web with Classic Blend™.

Download FREE evaluation software!
www.appliedreasoning.com

Copyright 1999 Applied Reasoning Systems Corporation. All rights reserved. Applied Reasoning, Classic Blend, the Classic Blend logo and the Applied Reasoning logo are all trademarks of Applied Reasoning Systems Corporation. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Applied Reasoning Systems Corporation is independent of Sun Microsystems, Inc. All other trademarks are the property of their respective owners.

APPLIEDREASONING™

www.appliedreasoning.com

800-260-2772

913-319-0900

When Worlds Collide

by Sean Gallagher
Editorial Director
sgallagher@fawcette.com

I never thought I'd see a day when Bill Gates parroted Scott McNealy, and Microsoft tried to restructure its development platform around an open standard. But both have happened in the last few months—Bill Gates talked about write once, run anywhere development tools, and the latest Microsoft vapor platform is based heavily on a rapidly emerging Internet standard. Talk about signs of an impending apocalypse.

Of course, neither of these involves Java—both center on XML. Then again, neither of these involves anything Microsoft will ship anytime soon. But it is ironic that, at this particular junction in history, the Microsoft and Java platform strategies would intersect in such an interesting way.

Whether we choose to admit it or not, much of the effort that has been put into the Java platform over the last two years has been to replicate the services and tools that Windows developers have had to work with. JavaServer Pages and JDBC even borrow parts of the names of Windows technologies. Others, like the Java Naming and Directory Interface (JNDI), match at some levels what Microsoft's platform offers—in JNDI's case, it's the new but widely unproven Active Directory Service. And while Microsoft loyalists may argue that the Java 2 Enterprise Edition (formally rolled out by Sun in December) is still mostly vapor, the same could easily be said about a great deal of Microsoft's DNA 2000 architecture.

While Java is definitely now a platform to be reckoned with, it doesn't mean that Java is ready to be all things to all people. The desktop is still the domain of Windows, and it's almost inevitable that any corporate application, written in Java or not, will find its way to or through a Windows environment in some form.

Religious platform fervor is for those without an application backlog or a ravenous user base—in other words, it's not for corporate developers. Still, living in a Windows world certainly doesn't mean that you have to live by Microsoft's rules—and working with Java doesn't mean you can't play with Bill Gates's toys when they fit the bill for a solution. In fact, it's possible to turn the tools of Microsoft's COM and DNA against themselves—as components of a (primarily) Java enterprise architecture. As two of this month's features show, you can effectively hijack the better parts of the Windows platform and turn them into resources for almost pure Java development.

In our cover story, Marcus Daley shows essentially how to hoist the Windows NT platform on its own petard—by turning Microsoft Transaction Server into a pure Java transaction processing server. Bill Block and Jerry Brunning describe bridging Java and COM objects through an XML interface, exposing data and methods trapped in the Windows world to Java developers.

As Microsoft continues to move its strategy more closely in line with that pioneered by Java, you can expect to be able to scavenge even more from the Windows world to help deliver Java-based solutions. Microsoft's XML strategy isn't necessarily a challenge to Java as much as it is an opportunity for Java developers to put Windows in its rightful place within the enterprise—as one operating system choice among many available in a multivendor environment.

Microsoft's XML strategy isn't necessarily a challenge to Java as much as it is an opportunity for Java developers to put Windows in its rightful place within the enterprise

Microsoft doesn't have to put a knife to its own throat for this to happen either—other vendors will certainly help open a vein or two. Lotus, for example, has just bundled in a COM interface to Domino. That doesn't seem like much of an entrée for Java. However, when it's combined with Lotus and IBM's support for Java through the Domino/WebSphere bundle, it provides a way to make back-end, Java-based services accessible to Windows clients through the COM interfaces that Windows developers are used to. And it provides Windows tools leverage. In fact, the Domino COM interface could make it possible for organizations to move away from the Microsoft platform on the server while keeping their investment in the Windows client intact.

Maybe this wasn't the kind of "embrace and extend" that Microsoft had in mind . . .

Go Online

Use the following DevX Locator+ code at www.java-pro.com to go directly to this related resource.

JP0002EN_T DevX Premier Club members can read this feature online.

Not a member yet? Go to www.DevX.com to sign up for the free Registered Level or to subscribe to the Premier Club.

Java™ 2 Platform Enterprise Edition

FOR THE REAL WORLD

Java™ 2 Platform Enterprise Edition

(J2EE) is the single most important standard to emerge since Web applications became the world's new way of doing business. Understanding this standard and how to implement it is critical to the success of your Web applications. Now you can learn what you need to know about the J2EE platform – and how to make it all work for you.

SEMINAR SCHEDULE

Chicago	Feb. 23
Detroit	Mar. 21
New York City	Feb. 1
Philadelphia	Feb. 23
Research Triangle Park, NC	Feb. 3
San Francisco	Mar. 7
Seattle	Mar. 8

FOR DETAILS:
www.silverstream.com/events

Attend a FREE half-day educational seminar *Java™ 2 Platform Enterprise Edition for the Real World*, brought to you by SilverStream Software and Sun Microsystems. Through technical demonstrations and instruction from industry experts, you'll gain a better understanding of the impact of the J2EE platform, the details of the specification and how you can use it to quickly deliver robust Web solutions.

WHO SHOULD ATTEND?

Developers and architects who know about J2EE, or are just starting to investigate it.

WHY ATTEND?

- Learn what you need to know about using the J2EE platform to implement real-world applications
- Discover how and when to use J2EE technologies such as Enterprise JavaBeans™ components (EJB), Servlets, JavaServer Pages™ components (JSP), Java™ Transaction Service (JTS) and Java™ Foundation Classes (including the GUI toolkit, JFC/Swing)
- Receive a Beta of the newest version of the SilverStream Application Server
- Get valuable materials to reference as you implement your applications
- Learn how to access additional resources such as training and services to help you get started

REGISTER TODAY FOR A FREE HALF-DAY EDUCATIONAL SEMINAR

www.silverstream.com

Or phone 1.888.823.9700 (toll-free in the U.S.) or 1.508.788.3260 (outside the U.S.)

"SilverStream raises Java ante."
— *PC Week*, November 1, 1999
Product review of SilverStream 3.0 Beta, which you will receive by attending the seminar.

LEARN HOW TO DELIVER REAL-WORLD APPLICATIONS WITH THE J2EE PLATFORM.

Copyright ©1999 SilverStream Software, Inc. SilverStream is a registered trademark and SilverStream Designer is a trademark of SilverStream Software, Inc. Sun, Sun Microsystems, the Sun logo, Java, Enterprise JavaBeans, and JavaServer Pages are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Sun microsystems | SilverStream®

JAVAPro

Jeff Miller, *Vice President, Publishing*

Editorial
Sean Gallagher, *Editorial Director*
Kay Keppler, *Executive Editor*
Janaya Reitz, *Managing Editor*

Contributors
Daniel F. Savarese, James W. Cooper,
Rick Grehan, Claude Duguay

Art & Production
Michael Hollister, *Art & Production Director*
Amy René Pelto, *Design Production Associate*
Christine Marquez, *Associate Production Director*
Kathleen Sweeney, *Associate Production Manager*
James Kattinge, *Senior Designer*
Kim Le, *Senior Designer*
James Kattinge, *Electronic Products Associate Art Director*
Robyn Pearson, *Electronic Production Associate*
Michael Furukawa, *Associate Production Manager*
Casey Mallough, *Associate Art Director*

Advertising Sales
Kelly Keane, *Regional Sales Manager*
Shana Deane, *Regional Sales Manager*
Tami Stanger, *Production Coordinator*

Circulation
Karen Koenen, *Circulation Director*
Katie Hathaway, *Associate Circulation Manager*
Fred Perry, *Associate Circulation Manager*
Evelyn Adenau, *Associate Circulation Manager*

Marketing
Anne Casanova, *Marketing Manager*
Libby Dickinson, *Marketing Communications Manager*
Margaret Horoszko, *Marketing Designer*

Conferences
Tena Carter, *General Manager*
Ann Odonnell, *Conference Manager*
Janet Brandt, *Conference Planner*
Robert Scoble, *Conference Planner*
Jennifer Brucker, *Conference Planner*
Michelle Grossman, *Conference Marketing Coordinator*
Stephen Gordon, *Director of Conference Sales*
Cedric Fellows, *Conference Sales Manager*

Electronic Products
Matthew Carter, *Editor & Producer*
Stefan Grunwedel, *Senior Editor*
Chris Hall, *Managing Editor*
Andrew King, *Associate Producer*
Sylvia Zele, *Associate Editor*
Diane Matcheck, *Product Reviews Editor*
Mana Tominaga, *Associate Editor*
Sherry Thomas-Zon, *Marketing Director*
Julianne Sturdivant, *Ancillary Products Manager*
Peter Walsh, *Product Manager*
Rachel Palmer, *Product Marketing Specialist*

Electronic Products, Advertising
Duane Dirstine, *National Sales Director*
Darrin Lord, *Regional Sales Manager*
Paul Seal, *Regional Sales Manager*
Amber Lynch, *Sales Representative*
Nathan Lenox, *Inside Sales Representative*
John Estalilla, *Southwest Sales Representative*
Kimberly Grandy, *Advertising Coordinator*

Customer Service
David Seymore, *Customer Service Manager*
Chris Ortiz, *Customer Service Representative*
Gerry Guzman, *Customer Service Representative*

Operations
Patricia Cameron-Loyd, *Chief Operations Officer*
Trisha Merriam, *Assistant to the Publisher*
Wendy Hartman-Carr, *Controller*
Aaron Wuele, *IS Manager*
Andy Chou, *Assistant IS Manager*
Andrew Nickels, *IT Specialist*
Pamela Davis, *HR/Benefits Administrator*
Lorenzo Valencia, *Facilities Manager*
Sam Azzouni, *A/R Accountant*
Keith Teague, *A/P Accountant*
Fina Pita, *Receptionist*

Fawcette Technical Publications
James E. Fawcette, *President*
Lynne Matthes, *Group Publisher*
Paul Spyksma, *Publisher, FTP Advertising Network*
Chris Kinsman, *Vice President, Technology*
Jeff Miller, *Vice President, Publishing*
Wilson, Sonsini, Goodrich & Rosati, *Corporate Counsel*

ONLINE RESOURCES

Registered Users
Locator+ code: **JP0002**
Type this code in the Locator+ field at the top of the *Java Pro* home page. You can find all the code in the current issue of *Java Pro* at www.java-pro.com. All the listings and associated files discussed in the articles are available free to Registered members in one ZIP file. To access code in issues from 1998 and 1999, type "JP" followed by the month and year. For example, the Locator+ code for the October 1998 issue is JP1098. To access code from issues published after December 1999, type "JP" followed by the year (two digits) and the month (two digits). For example, the Locator+ code for the January 2000 issue is JP0001.

Premier Club Members
Members of the Premier Club of the Development Exchange (www.devx.com, $24.95 for six months, $44.95 for one year) can get each article's listings and associated files in separate ZIP files. For example, to download the individual code file associated with the October 1998 "Ask the Java Pro" column by Daniel F. Savarese, you would type in JP1098AP as the Locator+ code. For any articles published after December 1999, however, the position of the year (two digits) and the month (two digits) will be switched in the Locator+ code. For example, to download the individual code file associated with the January 2000 "Ask the Java Pro" column, you would type in JP0001AP. Premier Club members can also access archives of all articles and related code ever published in *Java Pro*.

Java Zone
The Java Zone (www.java-zone.com) on the Development Exchange (www.devx.com) offers additional Java resources including product reviews, programming tips, weekly downloads, and discussion forums.

HOW TO REACH US

Editorial Offices:
Fawcette Technical Publications
209 Hamilton Avenue
Palo Alto, CA 94301-2500
Phone: 650-833-7100
Fax: 650-853-0230
Web: www.java-pro.com
E-mail: java-pro@fawcette.com

Article Submission
To submit articles for publication please contact:
Kay Keppler, Executive Editor, *Java Pro*
E-mail: kkeppler@fawcette.com
Download the Author Guidelines at www.java-pro.com/authguid.asp

Product Reviews
To submit products for review please contact:
Diane Matcheck, Product Reviews Editor
E-mail: dmatcheck@fawcette.com

Product Announcements
Please e-mail press releases to:
pressreleases@fawcette.com

Reprints and Permissions
For all *Java Pro* editorial and advertising reprints contact:
PARS International Corp.
114 East 32nd Street, Suite 503
New York, NY 10016
Phone: 212-779-4469
Fax: 212-779-4277

To quote from an article, please contact:
Tami Stanger
209 Hamilton Avenue
Palo Alto, CA 94301-2500
Phone: 650-833-7118
E-mail: tstanger@fawcette.com
Specify the issue date and title of the article, the portion you would like to quote, and the purpose.

Photocopy Rights:
Permission to photocopy for internal or personal use may be granted by Fawcette Technical Publications. Please contact Tami Stanger for more information.

Customer Service and Subscription Information
For subscription orders, inquiries, or address changes please contact Customer Service:
Java Pro
P.O. Box 54584
Boulder, CO 80322-4584
Phone: 303-945-5282
Fax: 650-321-3818
Foreign and Canadian orders must be payable in U.S. dollars, plus postage. The surface rate to Canada and Mexico is $52.97 per year. For all other countries the air mail rate is $78.97 per year.

Back Issues
To order Java Pro back issues, call 650-833-7100 or 800-848-5523 and ask for Customer Service. Back issues cost $10. Additional postage will be charged to deliveries outside the USA.

Reviews

Tango 2000
JFC Suite 2.1

Tango 2000

Developing by Numbers

by Michiel de Bruijn

Quick Facts:

Tango 2000 is a capable visual development environment for data-driven Web sites, but will probably not live up to the expectations of Java developers used to more generic and open tools.

Pervasive Software
12365 Riata Trace Pkwy., Bldg. II
Austin, TX 78727
Phone: 800-287-4383; 512-231-6000
Web: www.pervasive.com
Price: $495

Pervasive Software's latest development environment, Tango 2000, scores fairly well on the buzzword-compliancy test: Whether you're looking for support for COM, JavaBeans, or XML, it's all in the box. Pervasive also promises that all this functionality is as easy to use as it will ever get.

Certainly *sounds* great—but before getting too excited, consider the target audience for the product: Web developers looking for a "drag-and-drop" solution to building complex Web sites. So, although JavaBeans *are* supported in Tango 2000, you're limited to using a scripting language to glue their functionality to other parts of the system. Which, although useful, is not quite what the average *Java Pro* reader looks for in a development environment.

Forget About Code

If you can accept that editing code is not something you'll do often—if at all—within the Tango editor, the product still has plenty to offer. In fact, you might find the visual approach Tango 2000 takes to application development pretty cool (see figure).

You start composing your Tango application by dragging items from the action toolbar on the left-hand side of the screen onto your workspace. All programming constructs are represented by icons, with action-specific properties accessible through property sheets. Some of the available actions are powerful: You can manipulate files on the host machine, send mail, or insert custom HTML code, which can include JavaScript code that uses variables from other parts of your program.

Tango 2000 puts a strong emphasis on database connectivity. By selecting a data source, the available databases and field will appear in a separate workspace, ready to be dragged onto the data components, such as the Search Builder, New Record Builder, or Direct DBMS component, which allows for manual construction of SQL statements. Transactions are fully supported, as well. One very useful aspect of Tango's data manipulation functions is that they have integrated support for XML, making it easy to communicate with other front-end or—especially—back-end processes that support this structured data standard.

Tango supports linking to COM and JavaBean components, although instantiation of objects tends to be slow. However, this feature works well, as long as the component supports the "assign value(s) to properties, invoke event, read properties" model.

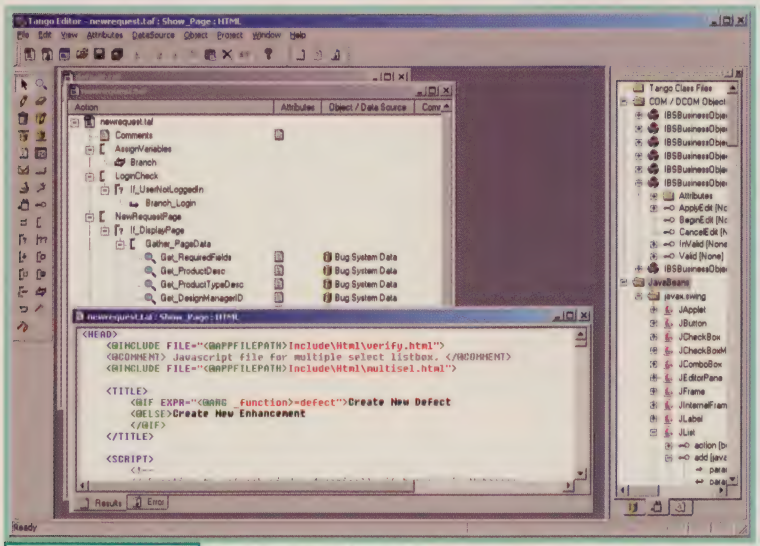
If you require more complex object interactions, you can write your own Java code (which you'll have to do outside the Tango environment, however) and execute it through something Pervasive calls its Java Action Server. This server module handles shifting data between the Tango code and your Java app, which needs to extend com.pvsw.tango.jas.Action and is obviously restricted in certain areas, such as its user interface.

Making Things Run

Deploying Tango applications requires the Tango Application server—a single-user copy is included with the development environment. Although the server components seemed usable and stable during testing, the relatively high cost of deployment combined with lack for real standards-based components may limit its attractiveness.

Other back-end components included with Tango 2000 are a Web analyzer, which provides detailed statistics on application usage, as well as a "workstation" copy of Pervasive.SQL. The latter is a fine relational database engine, but probably not a very exciting alternative for shops that have already invested in any other kind of ODBC-compliant database, especially since it seems to offer no specific functionality or performance enhancements when used together with the Tango editor.

Should you consider using Tango 2000 for your projects? If you're looking for a generic Java-flavored development environment that can be used to deploy to standards-based application servers, probably not. If, however, you must rapidly develop data-driven Web sites, especially using staff that has no extensive "real" programming experience, and you don't mind the product's proprietary structure, Tango may be ideal for you. ■



Cool Functionality Tango 2000's visual development environment enables you to build complex Web sites quickly using simple drag-and-drop operations.

Michiel de Bruijn lives in Rotterdam, The Netherlands, and spends his days as a networking and development specialist for an international media group. He seems to enjoy living on the bleeding edge of technology. Michiel always welcomes questions and comments: reach him at mdb@x42.net.

ease-of-use
leading-edge
time-to-market
server-side

WebLogic.
The one-word solution to your hyphenated needs.

java-based
mission-critical
best-of-breed
enterprise-scale
high-performance
time-to-market

Are Web applications part of your IT strategy?

Go to www.beasys.com and learn why the BEA WebLogic application server has been touted by the industry as the leading solution for quickly deploying and managing your high-performance, mission-critical, Java-based, enterprise-scale (etc.) Web applications.

Java Report 4 Cup Rating

BEA WebLogic®
www.beasys.com

© 1999 BEA Systems, Inc. All rights reserved. BEA, the BEA logo, and WebLogic are registered trademarks, and the BEA logo is a trademark of BEA Systems, Inc. Java is a trademark or registered trademark of Sun Microsystems, Inc. Other trademarks are the property of their respective owners.

bea™

JFC Suite 2.1

How Suite It Is!

by John Pearson

Quick Facts:

JFC Suite 2.1 enhances original JFC components, particularly JFCDataTable, JFCDataExplorer, and JFCDataCalendar. However, it is a bit pricey for small shops.

ProtoView Development Corp.
2540 Route 130
Cranbury, NJ 08512
Phone: 800-231-8588
Fax: 609-655-5353
Web: www.protoview.com
Price: $995 ($1,495 with Source Code)

Those who attended JavaOne in 1998 heard the announcement of the Java Foundation Classes (JFC). At the same conference, ProtoView announced the release of JFC Suite, its component suite to complement JFC. Now, more than a year later, Java continues to improve and develop. So, too, has the JFC Suite.

JFC Suite includes five major components: JFCDataTable, JFCDataExplorer, JFCDataInput, JFCDataTree, and JFCDataCalendar (see figure). All of these are all built on the Java Foundation Classes. Several of them are enhanced drop-in replacements for existing JFC components (for example, JFCDataTable is a drop-in replacement for JTree). All of the components are significant improvements over the originals.

JFCDataTable is an enhanced version of JTable. The first item that caught my attention was that you can print the entire table as well as selected rows. A print dialog and print preview dialogs are included. JFCDataTable has key mappings for several keyboard events (notably ESC, INSERT, and DELETE). For those who need more than a simple table, individual cells can contain other components, notably the JFCDataInput components. In addition, this component has other useful features such as sorting by columns and text searches (whole, string, or partial text). Because JFCDataTable is also a descendant of JPanel, it has other capabilities, such as notably enhanced colors, borders, and shadowing.

JFCDataTree, a descendant of JTree, is the next major component. It enhances the appearance of your trees by allowing more images per node (as many as six), customized indentation, customized coloring, dynamic node appearance, and drag-and-drop capabilities.

JFC DataExplorer is also a descendant of JTree, but it allows something JTree doesn't: You can drill down on data structures and display related data in the right-hand pane. You can add a grid, panel, or other components to make this a good way to organize and view data. In my genealogy application, I worked a bit on this to show family groups.

JFCDataInput is a library that contains a handful of data input components to make data validation easier. The components include JPVCurrency, JPVEdit (a regular edit box with some fun visual capabilities), JPVMask (a masked edit box), JPVNumeric (great for finances!), JPVTime (add this component and your application displays the time), and JPVPassword. For those who do a lot of work with dates, as I do, the DateEdit and DateEditLong components will be helpful. Additionally, a couple of other nifty components provide enhanced buttons, spinners, and labels.

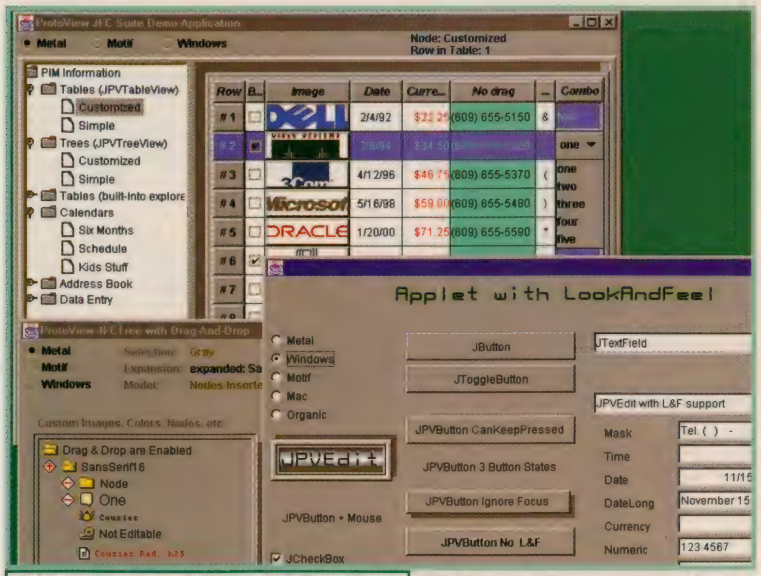
I have to admit, though, that my favorite components are part of the JFCDataCalendar. My hobby "program to end all programs" involves genealogy, so JFCDataCalendar has a lot to offer me. First of all, it's a calendar. It will show most of the dates you'll want to show (100 A.D. to 9999 A.D.), it allows for popup and pulldown dates, and it enables you to customize the contents of each cell. In addition to the generic calendar, it has two date selection components (JPVDatePlus and JPVDateLongPlus) that allow you to select dates from popup calendars.

In future versions, I would suggest that ProtoView add some sort of graphing capability—perhaps a 2D or 3D graph, a histogram (bar graph), and a way to point a couple of variables against an *x*- and *y*-axis. This would add to the product's appeal and make the price tag easier to swallow for small developers.

JFC Suite requires that the JDK 1.15 or JDK 1.2 be installed with JFC. However, ProtoView's documentation suggests that there may be a few anomalies with 1.15, so they recommend 1.2. On my NT 4 system, it took about 9MB to install.

The online documentation is well organized and helpful. The sample programs and applets that are included are equally important for learning how to use these products. Live, in-person tech support is also a part of ProtoView's products. I have called them a few times and received accurate help very quickly.

ProtoView's JFC Suite 2.1 is a component library custom designed for programmers who want to add a touch of professionalism and a lot of user friendliness to their programs. The components work well, are not too large to be included in a small application, and are frequently just what you need. ■



Tables, Trees, and Validated Data Entry JFC Suite 2.1 has something for everyone who is interested in making programs more user friendly and data entry more accurate.

John Pearson develops software for Java and Web development tools. He has contributed to several magazines including *Visual Basic Programmer's Journal* and *Visual Developer*.

It's A No Brainer.

Finally, An eCRM Solution With The Smarts To Be Java-based.

Now there's an alternative. YOUcentric, adapt-to-order eCRM.
Exact fit. Less time. Less money. Useable in real life.

Only YOUcentric Delivers A Platform Independent Solution.

Think About It.

youcentric™
ADAPT-TO-ORDER RELATIONSHIP MANAGEMENT

For more information call 888-462-6748, visit us at www.youcentric.com/eCRM, or e-mail us at javapro@youcentric.com

©2000 YOUcentric, Inc. All rights reserved. YOUcentric and the associated logo are trademarks and servicemarks of YOUcentric, Inc.

New Products

Database/ Middleware Tools

Developer Tools

Authoring/ Deployment Tools

Database/ MiddlewareTools

CryptoAdmin 4.0

This client-server application enables system administrators to initialize CryptoCard RB-1 Challenge-Response Tokens and centrally administer user databases. CryptoAdmin 4.0 is designed to work in conjunction with third-party products that have built-in support for CryptoCard tokens. The server updates the user database in real time, so you don't need to shut down and restart the the third-party application, and is compatible with Windows 95/NT, Solaris, Linux, and BSDi platforms. The client component runs on Win 32 platforms and features a simple GUI to ease issuing and management of CryptoCard RB-1 tokens. The CryptoAdmin client has a direct interface to the CryptoCard RB Initializer. Contact for pricing.

CryptoCard
300 March Rd., Ste. 304
Kanata, Ontario
Canada
K2K 2E2
Phone: 613-599-2441
Fax: 613-599-2442
E-mail: info@cryptocard.com
Web: www.Cryptocard.com

Developer Tools

Eureka:Intelligence

Eureka:Intelligence is a Web-based, integrated query, analysis, and reporting tool that allows you to query, manipulate, and format data for personal or shared use. Eureka:Intelligence features a 100% Pure Java client and server design to provide zero-desktop administration and cross-platform support. The client code is maintained and deployed automatically by the Eureka:Intelligence server. $595/user.

Sterling Software
7905 Golden Triangle Dr.
Eden Prairie, MN 55344-7227
Phone: 612-833-3700
Fax: 612-833-3701
Web: www.sterling.com

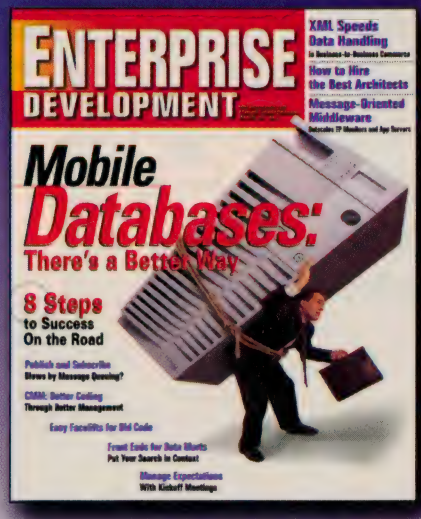
HSFormula

HSFormula is a formula engine/software component that gets string expressions and evaluates them at run time. Number, string, logical, and array types are handled in your expressions. Features include a rich operator

FREE SUBSCRIPTION

Enterprise Development

SUBSCRIBE ONLINE www.enterprisedev.com


ENTERPRISE DEVELOPMENT
XML Speeds Data Handling
How to Hire the Best Architects
Message-Oriented Middleware
Mobile Databases: There's a Better Way
8 Steps to Success On the Road

Architect enterprise apps and better manage the development process

Until now, no single magazine has met the needs of enterprise development managers. They've had to scour countless publications aimed at CIOs or coders in search of solutions to their specific problems.

Enterprise Development cuts through all the confusion to bring the right answers together under one cover. **And it's FREE to qualified subscribers.**

SUBSCRIBE ONLINE
www.enterprisedev.com

Or, complete the subscription form on the facing page
(fold, tape, and mail)

Or, fax your completed form to 650-321-3818

FTP FAWCETTE TECHNICAL PUBLICATIONS
209 Hamilton Avenue
Palo Alto, CA 94301-2500
Phone: 650.833.7100

* Publisher reserves the right to determine qualification for free subscriptions.

ENTERPRISE DEVELOPMENT

RECEIVE YOUR FREE SUBSCRIPTION

to **ENTERPRISE DEVELOPMENT**—the essential tool for the management and architecture of application development projects.

I wish to receive ENTERPRISE DEVELOPMENT Magazine free of charge! ☐ Yes! ❑ No

NAME		**TITLE**	
SIGNATURE (REQUIRED)	X	**COMPANY**	
DATE (REQUIRED)	/ /	**DEPARTMENT**	
PHONE	( )	**ADDRESS**	
FAX	( )	**CITY, STATE**	
E-MAIL ADDRESS		**ZIP/POSTAL CODE**	**COUNTRY**

Annual subscription rate is $99.95 for non-qualified subscribers.

PLEASE ANSWER ALL QUESTIONS, SIGN AND DATE THE CARD. INCOMPLETE FORMS CANNOT BE PROCESSED.

1. PRIMARY BUSINESS ACTIVITY AT YOUR LOCATION (SELECT ONLY ONE):

101 ❑ Finance, banking, accounting, insurance
102 ❑ Manufacturing other than computer hardware and software
103 ❑ Manufacturing computers and computer software
104 ❑ Non-computer retailer, distributor, wholesaler
105 ❑ Applications software development
106 ❑ Aerospace
107 ❑ Government, military
108 ❑ Agriculture, mining, construction, oil & gas
109 ❑ Transportation, utilities
110 ❑ Media, marketing, advertising
111 ❑ Real estate
112 ❑ Research and development
113 ❑ Engineering
114 ❑ VAR/Systems Integrators
115 ❑ Communications services
116 ❑ Computer-related retailer, distributor, wholesaler
117 ❑ Systems software development
118 ❑ Education
119 ❑ Health, medical, legal
120 ❑ Consulting (please specify)

_____

199 ❑ Other (please specify)

_____

2. WHAT IS YOUR CURRENT JOB TITLE? (SELECT ONLY ONE FROM THE FOLLOWING SIX (6) CATEGORIES):

Application development manager/architect:
201 ❑ Applications development manager
202 ❑ Software development director
203 ❑ Software architect
204 ❑ Software project leader
205 ❑ Project manager
206 ❑ Advanced technology evaluator
207 ❑ Software team leader
208 ❑ Software engineering supervisor
209 ❑ Systems analyst/designer
210 ❑ Database designer
211 ❑ Application manager/ administrator
295 ❑ Other (please specify)

_____

IS manager:
230 ❑ CIO, Technical VP
231 ❑ IS/MIS manager
296 ❑ Other (please specify)

_____

Application developer:
240 ❑ Applications developer/programmer
297 ❑ Other (please specify)

_____

Computer operations staff:
250 ❑ System Administrator
251 ❑ LAN Administrator
252 ❑ Help Desk
298 ❑ Other (please specify)

_____

Corporate manager:
260 ❑ CEO, president, partner, chairman, owner
261 ❑ VP, controller, director of marketing/sales, General manager, division/department head

Other (categories not listed above):
299 ❑ Other (please specify)

_____

3. ARE YOU DEVELOPING APPLICATIONS THAT INVOLVE WINDOWS NT/2000 OR DO YOU PLAN TO BE IN THE NEXT 12 MONTHS?

301 ❑ Yes 302 ❑ No

4. ARE YOU DEVELOPING APPLICATIONS THAT WILL INVOLVE ONE OR MORE SERVERS OR HOSTS OR DO YOU PLAN TO BE IN THE NEXT 12 MONTHS?

401 ❑ Yes 402 ❑ No

5. DO YOU EVALUATE, SPECIFY, RECOMMEND, AUTHORIZE, OR APPROVE THE PURCHASE OF (PLEASE CHECK ALL THAT APPLY):

501 ❑ Application development tools?
502 ❑ Database products?
503 ❑ Internet and/or Intranet products?
504 ❑ Middleware products?
505 ❑ Hardware servers?
506 ❑ Modeling tools
507 ❑ Testing/QA tools
508 ❑ Software process tools (Project/Process Management, Metrics, CM, etc.)
599 ❑ None of the above

6. WHICH OF THE FOLLOWING SOFTWARE SERVERS DO YOU BUY, SPECIFY, RECOMMEND, OR APPROVE THE PURCHASE OF AND/OR CURRENTLY HAVE INSTALLED AT THIS LOCATION (PLEASE CHECK ALL THAT APPLY):

	Buy, Specify, Recommend, Approve	Installed
Web	601 ❑	602 ❑
Database	603 ❑	604 ❑
Application	605 ❑	606 ❑
Transaction	607 ❑	608 ❑
E-mail	609 ❑	610 ❑
Imaging	611 ❑	612 ❑
Commerce	613 ❑	614 ❑
Remote Access	615 ❑	616 ❑
Communication/SNA	617 ❑	618 ❑

699 ❑ None of the above

For faster service fax both sides of this form to: 1-650-321-3818
OR fill out the electronic form at www.enterprisedev.com

4845

7. SERVER/HOST OPERATING SYSTEMS IN USE AT THIS ENTIRE LOCATION (PLEASE CHECK ALL THAT APPLY):

701 ❑ Windows NT/2000 Server
702 ❑ Unix (all flavors)
703 ❑ Netware/Intranetware
704 ❑ OS/400
705 ❑ VMS
706 ❑ MVS/VM
707 ❑ OS/2
799 ❑ Other (please specify)

_____

8. WHAT IS THE ANNUAL DOLLAR VALUE OF COMPUTER PRODUCTS AND SERVICES FOR WHICH YOU ARE PERSONALLY INVOLVED IN THE PURCHASING DECISIONS FOR YOUR ORGANIZATION'S OR LARGEST CLIENT'S INTERNAL USE?

801 ❑ More than $10 million
802 ❑ $5 million - $9,999,999
803 ❑ $2 million - $4,999,999
804 ❑ $1 million - $1,999,999
805 ❑ $500,000 - $999,999
806 ❑ $100,000 - $499,999
807 ❑ $50,000 - $99,999
808 ❑ $10,000 - $49,999
809 ❑ Less than $10,000

9. WHICH OF THE FOLLOWING CATEGORIES OF PRODUCTS DO YOU RECOMMEND, EVALUATE, SPECIFY, BUY, OR APPROVE THE PURCHASE OF? (CHECK ALL THAT APPLY):

901 ❑ Back office suites
902 ❑ N-Tier Capable Development Tools
903 ❑ ORBs
904 ❑ RPC middleware
905 ❑ Message Oriented Middleware
906 ❑ Application Integration Software
907 ❑ TP Monitors or Transaction Servers
908 ❑ Application Servers
909 ❑ Application Management Software
910 ❑ Component software (VBX, OCX, ActiveX, custom controls, JavaBeans, etc.)
911 ❑ Configuration Management/ Version Control
912 ❑ Project Management Tools
913 ❑ Process Management Tools
914 ❑ Estimation/planning/metrics tools
915 ❑ Class Libraries
916 ❑ Application Frameworks
917 ❑ C/C++ Tools
918 ❑ Java Tools
919 ❑ VB Tools
920 ❑ 4GL Tools
921 ❑ OLAP, Data Mining or Data Warehousing tools
922 ❑ Reporting Tools
923 ❑ Web Authoring Tools
924 ❑ Expert systems/rules engines
925 ❑ Repository
926 ❑ Security Tools/Firewall Software
998 ❑ Other (please specify)

_____

999 ❑ None of the above

10. DO YOU RECOMMEND, SPECIFY, APPROVE OR PURCHASE APPLICATION PACKAGES SUCH AS THOSE SOLD BY ORACLE, SAP, PEOPLESOFT, AND BAAN?

98 ❑ Yes 99 ❑ No

If YES, which types?
(Check ALL that apply):

101 ❑ ERP Suites
102 ❑ Human Resources
103 ❑ Sales Automation
104 ❑ Financial/Accounting
105 ❑ Retail/Point of Purchase
106 ❑ Manufacturing/Inventory
107 ❑ Supply Chain Management
199 ❑ Other (please specify)

_____

11. DO YOU RECOMMEND, SPECIFY, APPROVE OR PURCHASE APPLICATION DEVELOPMENT TOOLS FOR CUSTOMIZATION OF AND INTEGRATION WITH ERP OR OTHER APPLICATION PACKAGES?

201 ❑ Yes 202 ❑ No

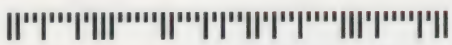
12. WHAT IS THE NUMBER OFEMPLOYEES OF THE LARGEST BUSINESS ENTITY BY WHOM YOU ARE EMPLOYED?

301 ❑ Under 100
302 ❑ 100 - 999
303 ❑ 1,000 - 4,999
304 ❑ 5,000 - 9,999
305 ❑ 10,000 - 49,999
306 ❑ More than 49,999

Publisher reserves the right to determine qualification for FREE subscriptions.

▼ FOLD AND TAPE, DO NOT STAPLE ▼

▼ FOLD AND TAPE, DO NOT STAPLE ▼



PLEASE MAKE SURE YOU'VE:
• Signed and dated the form
• Filled out the form completely
• Fax to: 1-650-321-3818 or fold, tape and mail (*please do not staple*)

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

ENTERPRISE DEVELOPMENT
PO BOX 469016
ESCONDIDO CA 92046-9619

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

set, such as numerical operators, and boolean operators. Array handling features include explicitly entering arrays. Variables and function calls are allowed and all are user (or programmer) defined. There is no limitation on variable and function names, and spaces are allowed. Variable and function names may be dynamic, and function argument count as well as types of arguments are all dynamic. Both Java and ActiveX versions are available. $200; $995 with source code.

HakanSoft
Web: www.hakansoft.com

Objectivity/DB 5.2

Objectivity/DB 5.2 is a high-end object database. New features include Active Schema for schema modification from any C++ program; Scalable Collections to create, build, and query collections that have millions of objects; Hot Failover to allow operation in a two-machine, hot-standby redundancy configuration; support for all Java native types from C++, plus Java 2, VisualWorks 5i, and Envy 5 for Smalltalk support. Platforms supported include Windows, Intel Linux, Solaris, AIX, Compaq Tru-64, HP-UX and SGI Irix. Languages supported include Java, C++, and Smalltalk. Contact for pricing.

Objectivity Inc.
301B East Evelyn Ave.
Mountain View, CA 94041
Phone: 800-767-6259; 650-254-7100
Fax: 650-254-7171
E-mail: info@objectivity.com
Web: www.objectivity.com

StarSQL Enterprise Edition 3.0

StarSQL Enterprise Edition offers direct access to IBM DB2 databases from any ODBC-compliant application. StarSQL Enterprise Edition works with either StarQuest's StarPipes or the Cisco Database Connection and runs on the desktop. Features include DRDA Level 3 over TCP/IP; stored procedures and static SQL support; ODBC V3 compliance; ADO and RDO support; password management; thread safe, standards-based compatibility; UNIX support; and Y2K compliance. Starting at $2,995.

StarQuest Software Inc.
1288 Ninth St.
Berkeley, CA 94710-1501
Phone: 510-528-2900; 800-763-0050
Fax: 510-528-2986
E-mail: sales@starquest.com
Web: www.starquest.com

Web Client 1.5

Web Client is a Java-based solution, giving GUI Internet/intranet access to any 3270-based application, such as CICS, IMS, TSO, UniKix. The Web client consists of several major components: a 3270 data stream capture tool; a visual editor that customizes and generates a GUI representation of the datastream; and a Java applet that displays the GUI representation in the user's browser. Free.

UniKix Technologies
8125 N. 23rd Ave., Ste. 195
Phoenix, AZ 85021
Phone: 800-765-2826
Fax: 800-724-1882
E-mail: marketing@unikix.com
Web: www.unikix.com

Authoring/Deployment Tools

InstallShield Java Edition 3.0

InstallShield Java Edition 3.0 (ISJE) provides developers with a visual tool designed to create a single Java installation that is consistent in look, feel, and functionality across multiple platforms. Developers can plug their own Java-based extensions into installations. The installation applet lets you create a seamless cross-platform launch site for your installation. ISJE 3.0 also enhances installation and deployment to the Solaris operating environment by leveraging Web Start wizard technology jointly developed by InstallShield and Sun Microsystems. Contact for pricing.

InstallShield Software Corp.
900 National Pkwy., Ste. 125
Schaumburg, IL 60173-5108
Phone: 847-240-9111; 800-374-4353;
Fax: 847-619-0788
Web: www.installshield.com

MOBILIZE YOUR APPLICATIONS

JUMPING BEANS

is a unique framework for creating Java applications which can seamlessly migrate from computer to computer while executing. Instead of deploying the different parts of a distributed application to different computers, a single program can handle all of the application logic. The advantages to you are:

Cut development costs by half: Your development team can concentrate on your business logic instead of managing distributed computing issues:

- Communication problems are gone because your team uses natural local programming instead of remote calls.
- Disconnected operation is transparently supported because Jumping Beans mobilizes the entire application, including code, state, data, resources, etc.
- Application security is made simple by utilizing the extensive security designed into Jumping Beans.
- Your application gains robustness because Jumping Beans transparently provides crash recovery and guaranteed delivery.
- Issues involving management of your distributed application are gone because Jumping Beans provides manageability through its central management console.

Cut your time to market by 1/3: You can initially deploy software with only minimal functionality, and transparently deploy upgrades and enhancements later. Upgrades can be made in stages because you can deploy different versions of your software simultaneously.

Reduce network connectivity and traffic: Send your application to the source of the data (such as remote databases, configuration files, users, etc), and disconnect. Your application can analyze the data locally and later report just the results when re-connected.

Don't re-invent what you already have: Jumping Beans does not require you to re-invent your existing software in order to take advantage of mobility. Instead, Jumping Beans will work alongside your existing software to create new features not possible without mobility.

"Jumping Beans allowes us to easily accomplish several critical goals. Our earlier applications used CORBA Inter ORB references, which limited our product's functionality. After using Ad Astra's Mobile CORBA, however, we could improve our applications mobility across all or our client bases."
- Suresh Neelagaru, M.D.
President of Medstream Software Labs

1999 JavaOne OOP '99 BEST OF SHOW Best Framework

1998 SIGS Conference JAVA Development BEST OF SHOW Best Java Frameworks

www.JumpingBeans.com
or call
408-738-4616

"Jumping Beans" and the Bouncing Bean Logo are trademarks of Ad Astra Engineering, Inc. Java, 100% Pure Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Transform MTS into a Pure Java

Overcome critical cross-platform limitations by RMI-enabling the Microsoft VM

Marcus Daley is a senior engineer at Intel. He has been involved with Java development since its birth. In 1995, while working with Utah State University, he pioneered research into 3D modeling applications written in Java and VRML. Later, he joined a startup GIS company where he developed Java-based intranet solutions for small businesses and local governments. Marcus lives in Portland, OR, and can be reached at marcus@mdaley.net.

When Java programmers decide to build a Java component that needs to be hosted in an object monitor, most do not consider Microsoft Transaction Server (MTS) as a hosting environment. But MTS can be a reasonable—perhaps sometimes the best—solution for this distributed computing problem.

The problem is this: MTS is Microsoft's proprietary object monitor, and it was built to work only with Component Object Model (COM) components, which are Microsoft's answer to distributed object computing. However, COM components can be built in any language that supports Microsoft's COM specification, and Java is one of those languages. Java components running on different operating systems, particularly those using non-Microsoft virtual machines (VMs), must speak COM or DCOM to use MTS. This drawback can be critical when you build cross-platform clients.

To eliminate this problem, thus enabling MTS to become a Java component-hosting environment, you must RMI-enable the Microsoft VM. Combining the technologies of Java's Remote Method Invocation (RMI) API and the Microsoft VM into a simple proxy class enables any pure Java VM that supports RMI to call Java components hosted inside MTS.

Remote Method Invocation

Ever since RMI became a core part of JDK version 1.1 in 1996, it has been a valuable mechanism for performing distributed computing between Java components. RMI is similar to many distributed technologies such as RPC,

Object Monitor

by Marcus Daley

CORBA, and DCOM, but it does not inherit the level of complexity that surrounds those technologies.

RMI basically conforms to client-server methodology. For example, when you use RMI, usually a client invokes Java methods hosted by a server. The client and server are thought of as separate machines located on a TCP/IP-based network. However, they can both be located on the same machine or even in the same component.

When creating an RMI server, download a Java development kit that comes with the rmic and rmiregistry tools (see Table 1). These tools are not part of the Microsoft SDK, so you will need to download another Java development kit (such as Sun's) that contains them. And, since Microsoft's VM does not support RMI out-of-the box, you will need to download either IBM's or Microsoft's RMI patch for supporting RMI in Microsoft's VM. These patches can be found at http://alphaworks.ibm.com/tech/RMI or ftp://ftp.microsoft.com/developr/MSDN/UnSup-ed/. Please carefully read the licensing agreements before using either of the RMI patches or the tools in the Sun JDK.

One way to create an RMI server is to build the Java classes that will make up the implementation of the RMI server first (see Figure 1):

Tool	Function
Rmiregistry	A naming lookup service for RMI. The rmiregistry can be thought of as an online hashtable that associates names with distributed object references.
Rmic	A tool that generates skeletons and/or stubs used by clients and servers to facilitate distributed communication.

Table 1 Useful Tools When creating an RMI server, download a Java development kit that comes with these two tools.

```
// Standard Java class that contains the
// business logic

public class NameServer
{
  public String getName (String name)
  {
    return name + " has done it!";
  }
}
```

Next, create interfaces that contain the method signatures that will be exposed to the clients of the RMI server. These interfaces need to extend java.rmi.Remote, and each interface method must declare that it throws a RemoteException:

```
// Standard interface that exposes the
// methods available in the RMI server

public interface NameServerRMIInterface
  extends java.rmi.Remote
{
  public String getNameImpl (String name)
    throws java.rmi.RemoteException;
}
```

One of MTS's most appealing features? It's free.

For those of you who crave design pattern terminology, think of these interfaces as bridges that create abstractions, allowing the method signatures to remain the same while allowing the implementation to vary based on the RMI server.

Either a Java IDE or Sun's and Microsoft's Java development kit will work for developing RMI-capable client and server classes. When making an RMI server component that is hosted inside MTS, it is important to remember that all MTS-hosted Java components require Microsoft's VM (currently Java 1.1.7 compliant). This

Listing 1 Classy Servers RMI server classes must extend a RemoteObject subclass. In this case, the NameServerProxy class acts as the RMI server.

```
import java.rmi.*;
import java.rmi.server.*;
import java.rmi.registry.*;
// Special Microsoft Packages //
import com.ms.com.*;          //
import com.ms.win32.*;        //
////////////////////////////////

// This class extends and inherits standard RMI classes
public class NameServerProxy extends UnicastRemoteObject
  implements NameServerRMIInterface, Runnable
{
  // The thread that will be used to create an active
  // RMI registry and bind the NameServerProxy class to
  // the registry
  public Thread rmiThread;

  // This method makes a COM call that starts the
  // registry thread
  public static void main (String args [])
  {
    // Retrieve an IUnknown pointer
    IUnknown nameServerProxy = Ole32.CoCreateInstance
      (Ole32.CLSIDFromProgID ("NameServerProxy.1"),
      null, 4 | 16, new _Guid (
      "{00020400-0000-0000-C000-000000000046}"));
    // Call a subroutine supported by the object referred to
    // by the IUnknown pointer
    Dispatch.callSub (nameServerProxy, "startServer");
    // Release the IUnknown pointer
    ComLib.release (nameServerProxy);
  }

  // The parameterless constructor that must throw a
  // RemoteException
  public NameServerProxy ()
    throws RemoteException
  {
  }

  // The proxy method that accepts RMI calls and marshalls
  // them to the actual method on the NameServer component
  // via a COM call
  public String getNameImpl (String name)
    throws java.rmi.RemoteException
  {
    // Retrieve an IUnknown pointer
    IUnknown nameserver = Ole32.CoCreateInstance
      (Ole32.CLSIDFromProgID ("NameServer.1"), null, 4 | 16,
       new _Guid ("{00020400-0000-0000-C000-000000000046}"));
    // Make the call to the appropriate component and method
    // and retrieve a return value
    Variant result = Dispatch.call (
      nameserver, "getName", name);
    // Release the IUnknown pointer
    ComLib.release (nameserver);

    // We know that the return value was a String so
    // return the String value contained in the Variant
    // return value
    return result.getString ();
  }

  // Start a thread that starts the private rmiregistry and
  // binds the NameServerProxy class to the RMI registry
  public void startServer ()
  {
    try {
      rmiThread = new Thread (new NameServerProxy ());
      rmiThread.start ();
    } catch (RemoteException re) {
      re.printStackTrace ();
    }
  }

  // The body of the thread
  public void run ()
  {
    try {
      // Create a private registry on the standard
      // registry port
      Registry reg = LocateRegistry.createRegistry
        (Registry.REGISTRY_PORT);
      NameServerProxy rmiProxy = new NameServerProxy ();
      // Bind an instance of the NameServerProxy to the
      // private RMI registry
      reg.rebind ("NameServerProxy", rmiProxy);
    } catch (Exception e) {
      e.printStackTrace ();
    }
  }
}
```

means that MTS-enabled Java components cannot use any Java 2-specific technology (such as weak references, HashMaps, or the new Java 2 RMI security policies) unless Microsoft starts supporting the Java 1.2 specification. However, the Java clients of MTS components can use the Java 1.2 specification if the client's VM is 1.2 compliant (such as Sun's 1.2.2 VM).

RMI server classes must extend a RemoteObject subclass, the most common of which is the UnicastRemoteObject class. Server classes must also implement the special RMI interfaces. The constructor of a server class that extends UnicastRemoteObject must declare that it throws a RemoteException. Such an exception will be thrown at construction time if the class cannot be created or if a stub is not available. In the sample code in Listing 1, the NameServerProxy class acts as the RMI server.

To facilitate the distributed communication process, two classes must be generated by the rmic tool. These class files are known as the stub and skeleton classes. Their purpose is to marshal data between the client and server components. If the client and server are to work, the client must use a stub and the server must use a skeleton. Rmic generates the stub and skeleton class files when it is executed against an RMI server class that contains a RemoteObject implementation. To use rmic with the sample NameServerProxy class, just use rmic NameServerProxy. Once rmic has generated the classes, they must be locatable by the RMI server's CLASSPATH, and the stub must be locatable by the client's CLASSPATH.

Once the server code has been finished and the skeleton has been generated, a server instance needs to be registered with the rmiregistry. Start the rmiregistry server from the command-line or, programmatically, have the server start its own private registry. The sample proxy code in Listing 1 creates a private RMI registry in the run() method. If you start an RMI registry server within your own code rather than from the command-line, the RMI process becomes more transparent and tightly integrated into the MTS process. It also eliminates the need for a third-party rmiregistry executable (such as Sun's rmiregistry server).

Since the client component does not require extension of a specific super class or implementation of a special RMI interface, it is easy to implement. First, the client locates the rmiregistry and asks for a remote object by name. A reference to the remote object is returned to the client. Then the client casts the reference to the appropriate interface. At this point, the client can call any method defined in the interface as though the remote object were a local class instance (see Listing 2), although the implementation may be located halfway around the world.

Microsoft Java Extensions and MTS

Microsoft has been part of the Java community since it licensed Java in December 1996. Since then, Microsoft has built high-performance VMs that

Free
SSL-C, SSL-J or S/MIME-C SDK.*

Get a Jump Start on Securing Your e-Business Applications.

Connect to the power of RSA BSAFE® e-security components.

It seems like the whole world is out there racing to record e-business profits. So where does that leave you? If you find yourself stuck curbside, while competitors leave you behind to sort through e-security issues, then RSA Security has the power boost you need – whether you develop apps for Web business, general client-server, remote management or embedded devices.

RSA BSAFE components. Mega volts of security. Our RSA BSAFE e-security components give you all the tools you need to power up your e-business applications. For SSL in C, SSL for Java™ technology and S/MIME, we get you rolling in a way those under-powered off-the-shelf browsers just can't match. In one jolt, you'll get full-strength crypto (available worldwide*), more extensive sample code and protocol links for services such as Telnet, FTP, NNTP, SMTP and POP3. And, by combining SSL and S/MIME security protocols with the industry's most trusted cryptography, RSA BSAFE security gives you the jump you need to get to market faster.

RSA BSAFE e-security. Power you control. Don't be stalled by e-security issues. With RSA BSAFE security components you're in charge of the implementation of our airtight security within your application. You control the interface, error recovery and the best balance of strength vs. accessibility for your needs. Best of all, you start off with the confidence that your e-security operates independently of changes to a browser. Your apps will always be completely interoperable.

RSA Security. Rev up with the best. Connect with RSA Security and you work with the industry leader. We've been the driving force behind putting SSL and S/MIME on the map as industry standard security protocols. No one knows more about how to build secure applications, so you can count on us to provide tools and support you can use to meet your goals on time and on budget.

Get a jump start on the competition. **Qualify today for a free RSA BSAFE SSL-C, SSL-J or S/MIME-C software development kit* and get charged up.**

1.800.782.5453 or www.rsasecurity.com/go/jumpstart

RSA SECURITY™

The Most Trusted Name in e-Security

RSA SECURED™

Products displaying this symbol contain the most trusted e-security technology

*Export or other restrictions may apply. RSA, BSAFE, the RSA Security and RSA Secured logos are trademarks or registered trademarks of RSA Security Inc. Java is a trademark of Sun Microsystems, Inc. © 1999 RSA Security Inc. All rights reserved.

Listing 2 Easy Clients The client component is easy to implement because it does not require an extension of a specific super class or implementation of a special RMI interface.

```
// Standard RMI client that retrieves an object reference
// and makes a method call

import java.rmi.*;

public class RMIClient
{
  public static void main (String args [])
  {
    try {
      NameServerRMIInterface nsri = (NameServerRMIInterface)
        Naming.lookup ("//yourmachinename/NameServerProxy");
      System.out.println (nsri.getNameImpl (
        "Someone's name"));
    } catch (Exception e) {
      e.printStackTrace ();
    }
  }
}
```

rival many other VM distributions. Microsoft has integrated Java into many parts of its operating system and enabled the language to work with many Microsoft-centric technologies such as COM and ActiveX. When Microsoft integrated these technologies into Java, it included several Windows-specific packages as a part of its VM. Some of these additions are well-known and range from the new Windows Foundation Classes (com.ms.wfc.*) to the Windows API integration contained in the com.ms.win32 package.

The Windows Foundation Classes are a new set of classes Microsoft provided to build high-performance Windows-only applications. These classes allow experienced Windows developers to leverage their Windows knowledge while using Java to build Windows applications. While these new classes are interesting, they are not thoroughly discussed in this article. Instead, this article focuses on the Ole32 class contained within the com.ms.win32 package.

If used wisely, Microsoft's enhancements can add value to certain Java projects

Microsoft has provided a small subset of the Win32 API in the com.ms.win32.Ole32 class. This subset opens the power of Microsoft COM programming to Java developers building applications for the Windows platform. Methods contained within this class should not be difficult for advanced Windows programmers who typically use C/C++ or Visual Basic to write Windows applications using similar function calls. The same is true for Java programmers wishing to include this class and its associated methods within their Java code.

The code accompanying this article is written to achieve interoperability with MTS. If you are using MTS to service multiple VMs (including Sun's), these enhancements become necessary. Fortunately, use of the special Microsoft packages in this article has been carefully limited to a single proxy class that marshals RMI calls into COM calls (this is explained later in this article). When you use any of the com.ms.win32 classes, please keep in mind that you are tying yourself to the Windows platform in one form or another. In fact, Microsoft's special packages are a source of controversy and have drawn media attention over the last few years. However, if these Java enhancements are used wisely, they can bring added value to certain Java projects.

Microsoft's answer to the middleware space is MTS. In other words, MTS is Microsoft's Transaction Processing Monitor (TPO). MTS is a powerful TPO because it can host many COM objects in single or multiple processes (called packages) and manage the security rights of each process. MTS also provides a way to create out-of-process COM objects or even remote DCOM server objects.

One of MTS's most appealing features is that it is free. Although other, more costly, application servers are available from vendors that support CORBA and the Enterprise JavaBean standard, MTS is a solid and reliable solution for those projects that do not require those features. MTS is included in the NT Option Pack for NT 4.x and has been integrated into the Windows 2000 operating system as part of the Windows 2000's Component Services (otherwise known as COM+). Windows 2000 Component Services and NT 4's MTS work exactly the same on basic components without any alteration (as of the RC2 release of Windows 2000). Therefore, unless specified, both will be referred to as MTS.

Creating Java COM components that work within MTS doesn't require using confusing Microsoft compiler directives. In fact, pure Java developers should encounter nothing unfamiliar in the process. It requires only that you in-

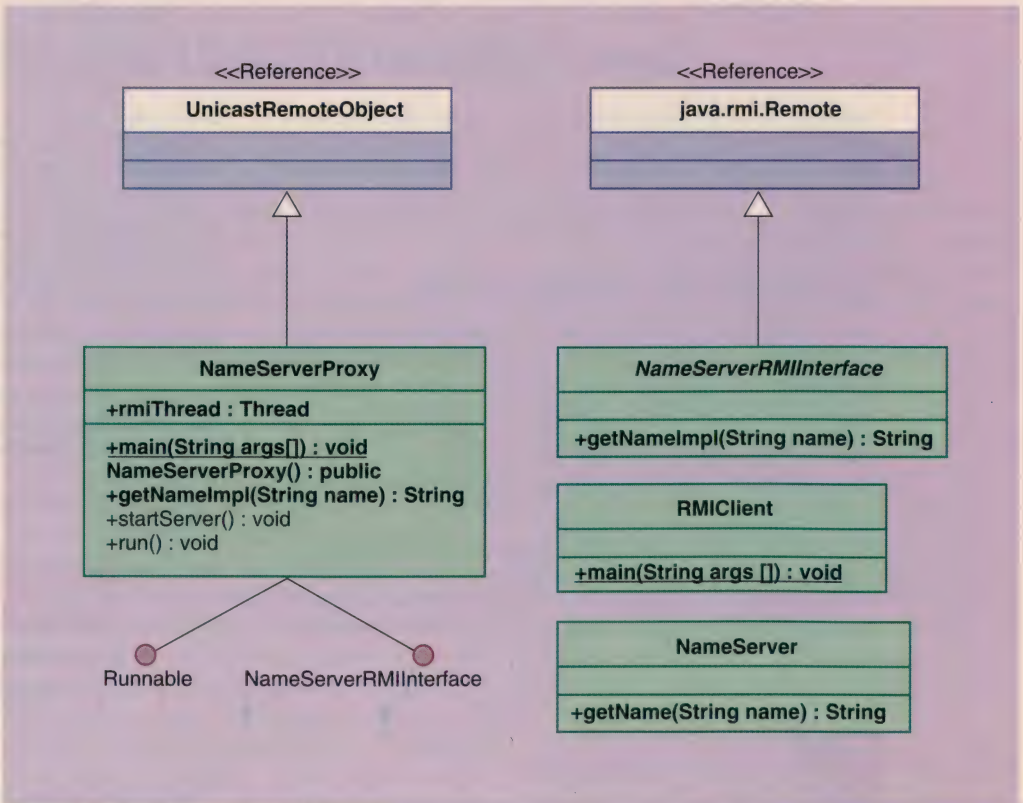


Figure 1 Class Relationships These are the class relationships in our RMI server.

Find hidden problems automatically with NuMega DevPartner™ Java™ Edition!

Serious problems can lurk quietly in your Java™ code, and if left undetected, can cause performance and reliability issues after your software is deployed. **NuMega DevPartner™ Java™ Edition*** from Compuware® automatically finds performance bottlenecks and hidden runtime errors in your Java code, which can save you days of valuable development time.

DevPartner Java Edition is a suite of developer productivity tools that accelerate software development and improve the performance and reliability of your Java code. DevPartner Java Edition includes **TrueTime** — the award-winning performance profiler that automatically pinpoints performance bottlenecks, with or without source code; **TrueCoverage** — the code coverage analyzer that automatically measures and tracks tested code, untested code and code base stability; and **JCheck** — the visual thread and event analyzer that automatically detects thread errors, runtime errors, leaks, unhandled exceptions and other potentially serious problems. Plus, all DevPartner Java Edition tools have powerful multi-VM switching capability, which can help you find behavioral problems that occur when your code runs in different virtual machines.

Find hidden problems in your Java code with NuMega DevPartner Java Edition, and deliver the fastest, most reliable Java applications and components with confidence!

Supports Java 2

Accelerating Software Development
with
NuMega DevPartner Java Edition from Compuware

NuMega DevPartner Java Edition includes:

NuMega TrueTime™ Java Edition — Performance Profiler
NuMega TrueCoverage™ Java Edition — Code Coverage Analyzer
NuMega JCheck™ — Thread and Event Analyzer

Call today to request your copy of the technical white paper
"Effective Unit Testing for Java Developers"
or download the paper now by visiting
www.compuware.com/numega

1-800-4-NUMEGA
www.compuware.com/numega

JDJ Java Developer's Journal World Class Award

NuMega DevPartner for Java "...Outstanding" 5 Cup Rating Java Report, August 1999

COMPUWARE
People and software for business applications™

30-Day Money Back Guarantee

Copyright © 2000 Compuware Corporation. All rights reserved. All trademarks and registered trademarks are the property of their respective owners.
Java and all Java-based marks are trademarks of Sun Microsystems, Inc. in the United States and other countries.

*Formerly DevPartner for Java™

clude a few Java packages from Microsoft. You will also need to compile the Java proxy with Microsoft's compiler.

To begin, download the latest Microsoft Java SDK and VM from www.microsoft.com/java/. Once you install these two downloads, you are ready to create a Java COM object that can be hosted in MTS (I would also suggest installing the RMI patch now).

First, examine the NameServerProxy code shown in Listing 1. Two packages, com.ms.com.* and com.ms.win32.*, need to be imported from the Microsoft class files. Those packages enable a Java class to perform standard COM calls as though it were a Microsoft C++ or Visual Basic program. The first package, com.ms.com.*, imports the standard COM classes that are necessary to perform basic COM calls from Java (such as the Variant data type, Dispatch class, and IUnknown class). The second package, com.ms.win32.*, imports classes that represent the standard Win32 API calls and the Ole32 class. The Ole32 class contains many of the Win32 API functions that are used to create and manipulate COM component instances.

The proxy class (Listing 1) shows two methods that use these special Microsoft classes. Explanations for each class and method usage can be seen in Table 2.

Class or method	Description
IUnknown	Similar to the interface that refers to an object reference returned from an RMI call. IUnknown represents an arbitrary COM object.
Ole32	A Java class containing Win32 API functions that create and manipulate COM component instances.
_Guid	Represents a globally unique identifier (GUID). A description of how Microsoft creates GUIDs can be seen at: www.microsoft.com/asf/spec3/c.htm.
Dispatch	Provides a mechanism for Java programs to invoke methods on a COM object represented by an IUnknown instance.
ComLib	Provides static methods that can be used to control a COM object instance.
Variant	A common data type used in COM communication. It represents a union of several different data types such as integers, floats, booleans, strings, and so on.
Ole32.CoCreateInstance()	Creates an instance of a COM object represented by a GUID and returns a specific interface supported by that COM object.
Ole32.CLSIDFromProgID()	COM objects can be identified by a GUID and a unique, user-friendly name. If you know the user-friendly name of the COM object, then you can get its corresponding GUID using this method.
Dispatch.callSub()	Invokes a function on a COM object that returns void.
ComLib.release()	Releases an instance of a COM object. This implies that you will no longer be using the COM object.
Dispatch.call()	Invokes a function on a COM object that returns a Variant.
Variant.getString()	Returns the String value contained in the Variant union.

Table 2 Imported Goods The Microsoft class file package com.ms.win32.* imports classes that represent the standard Win 32 API calls and the Ole32 class.

Advanced JDBC Programming
JAVA Pro
REVIEWS
Improving
Database
Queries
Metadata Gives You Reusable Code
Save 61% off the cover price!
Subscribe today, and you'll pay the low rate of $29.95 for a full year–13 issues in all. That's a savings of more than 61% off the newsstand rate!
One year is thirteen issues. Annual cover price is $71.40.
Building Serious Java Apps?
The professional developer's independent source for information about building serious Java applications.
Java Pro takes you beyond the hype by delivering hands-on, code-intensive, product-oriented information that helps you harness Java's unique capabilities in your enterprise-wide development efforts. Each month, Java Pro tackles tough development issues and provides unbiased, no-nonsense coverage of the latest tools on the market.
Order online at:
www.java-pro.com
Or, call 1-800-848-5523
Outside the U.S. call 650-833-7100
Money-back guarantee
If you are not satisfied with Java Pro at any time, we will refund the cost of your subscription.
No questions asked.
FTP FAWCETTE TECHNICAL PUBLICATIONS
Java is a trademark of Sun Microsystems, Inc.

Become a Subscriber NOW!

JAVA™ Pro

Yes! ☑ Send me one year (13 issues) of *JAVA Pro* for just $29.95

Name: _____

Address: _____

City/ State/ Zip: _____

Country: _____ E-mail: _____

❑ Bill Me ❑ Payment Enclosed Charge my: ❑ Visa ❑ MC ❑ AMEX

Card Number: _____ Exp. _____

Signature: _____

One year is 12 issues plus 1 special issue. Annual cover price is $77.35. Make checks payable to JavaPro. Please allow 4-6 weeks for delivery of first issue. International subscription must be payable in U.S. Dollars plus postage. Canada/Mexico add $18/year; all other countries add $44/year for air mail. Canadian GST included. Java is a trademark of Sun Microsystems.

11/99

4112

Become a Subscriber NOW!

JAVA™ Pro

Yes! ☑ Send me one year (13 issues) of *JAVA Pro* for just $29.95

Name: _____

Address: _____

City/ State/ Zip: _____

Country: _____ E-mail: _____

❑ Bill Me ❑ Payment Enclosed Charge my: ❑ Visa ❑ MC ❑ AMEX

Card Number: _____ Exp. _____

Signature: _____

One year is 12 issues plus 1 special issue. Annual cover price is $77.35. Make checks payable to JavaPro. Please allow 4-6 weeks for delivery of first issue. International subscription must be payable in U.S. Dollars plus postage. Canada/Mexico add $18/year; all other countries add $44/year for air mail. Canadian GST included. Java is a trademark of Sun Microsystems.

11/99

4112

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVA™ Pro
PO BOX 54584
BOULDER CO 80323-4584

Subscribe
online at
http://www.java-pro.com

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVA™ Pro
PO BOX 54584
BOULDER CO 80323-4584

Subscribe
online at
http://www.java-pro.com

The first call to CoCreateInstance() method in the NameServerProxy class is found in the main method. CoCreateInstance()creates a COM instance of the NameServerProxy class by requesting it as NameServerProxy.1, its COM friendly name. The lengthy String value passed as the last parameter of the method is the String representation of a GUID. The GUID represents the IDispatch interface that is part of the NameServerProxy object. Once the COM component is created, the Dispatch class calls a subroutine on the COM object named startServer(). The startServer() method starts a Java thread in its own private RMI registry and registers the NameServerProxy class as an available RMI server. Finally, the ComLib class calls the release method by passing the IUnknown NameServerProxy variable as the only parameter. This causes the NameServerProxy Java COM object to be released, since it will no longer be used.

The getNameImpl() method of the NameServerProxy class acts as a surrogate method for the actual getName() method found in the standard Java NameServer class. Java classes using RMI that invoke a method found in the Java NameServer class call this method. The NameServerProxy class accepts these RMI calls and marshals them to the NameServer class by performing a COM call.

One thing to note in the getNameImpl() method is that the return type from the standard Java method in the NameServer class has become a Variant data type. This happened because all COM calls using IDispatch require parameters and return types to be supported in the Variant union. Since all of the Dispatch class methods return a Variant, you can capture any data type returned from a Java method that was invoked via COM as long as the return type is supported in the Variant union.

Any Java 1.1 Java compiler can compile the NameServer class. However, the NameServerProxy class must be compiled using Microsoft's JVC compiler found in its 3.2 SDK (this is because of the two Microsoft-specific packages). But how did the standard Java 1.1 class NameServer became a COM object without special recompilation or Microsoft extensions? And how did the NameServerProxy object became a COM object without becoming a DLL or requiring an IDL file? These things are taken care of by javareg, provided by Microsoft, which turns standard Java 1.1 classes into COM objects. This tool can assign a GUID, user-friendly name, and other interesting capabilities to the Java class. These features are supported by javareg's command-line options, for example: javareg /class:NameServer /register /progid: NameServer.1. This tool is located in Microsoft 3.2 SDK /bin directory.

That's how a Java class can become a COM component without recompiling or comprising its compatibility. After compiling the NameServerProxy class with the Microsoft JVC compiler, it also must be javareged in the same way that the NameServer class was. Doing this will also make the NameServerProxy class a COM object.

When using the javareg tool and the Microsoft VM, the Java classes become IDispatch-capable COM objects and magically inherit the IDispatch interface. The IDispatch interface is represented by the GUID seen in Listing 1. The IDispatch interface allows clients of a COM object to query the object to see what public methods it contains. In the getNameImpl() method, I assumed there was a method called getName() that required a single String parameter and would return a data type supported by the Variant union. If such a method signature had not existed, an exception would have been thrown and the code would have required a change. The code would be fixed to include a call to an existing method signature.

The IDispatch COM interface is the simplest way to use Microsoft's COM architecture. It is

SQL
shouldn't mean
Seriously
Questionable Listings.
We know what great database jobs are all about. We do our homework and consider your needs and the employer's. We deliver fresh, focused opportunities from start-ups to Fortune 500 companies. And we e-mail ideal jobs straight to your inbox—for free.
Sign up with Career Central for Database Developers.
www.careercentral.com/database
career central®
for database developers

also the easiest way to achieve the RMI-over-COM marshalling required by MTS when non-Microsoft VMs are used.

Putting it all Together

So, how do we execute the sample NameServer class files? First, ensure that the NT Option Pack and the latest NT Service Pack are installed on your machine. An option in the NT Option Pack installation is to install MTS. (Select the MTS installation checkbox.) If you use Windows 2000, MTS is already installed in the form of the Component Services included in the operating system.

You will need to install the latest Microsoft SDK and VM, apply the RMI patch, and install a non-Microsoft VM that supports RMI (such as the Sun JDK 1.2.2). Installing a VM other than Microsoft's allows you to test the interoperability of multiple and differing VMs calling RMI methods on a COM object hosted inside of MTS.

The NameServerProxy class should be compiled using the Microsoft Java compiler (JVC). The other classes and interface can be compiled by any Java compiler you choose. These classes must be locatable in each VM's CLASSPATH, so make appropriate changes, if needed.

The javareg tool registers both the NameServer class and the NameServerProxy class and assigns them friendly names. Friendly names are called programmatic identifiers (progid) and are unique name mappings to a specific GUID. For convenience, I would suggest using the same friendly names as those included in the NameServerProxy code.

Once that is done, the newly registered Java COM objects need to be hosted within MTS, which is handled through the Microsoft Transaction Console (MTC) or Component Services Management Console (CSMC). Choose which tool you want, open it, double-click on the Computers icon, and double-click on the My Computer icon. This is where the two tools diverge. If you use the MTC (Windows NT 4.x), right-click on the Packages Installed icon and click New then Package. A dialog box will appear asking you what type of package you want to create. Click Create Empty Package and type the name of the package into the dialog box that appears (the name can be anything). Then, another dialog box appears that allows you to set security attributes on the package. Bypass this box by clicking Finish. The empty package is created.

If you use the CSMC (Windows 2000), follow the same steps with a few exceptions. The CSMC names are different, and you'll get a few additional dialog boxes. Right-click on the COM+ Applications icon and click New then Application. From there a dialog box will appear. Click Next, then click Create Empty Package and type the package name into the dialog box. Click Next (leaving the rest of the choices at their default state), then Next again, then Finish. This creates an empty package that will act as a host for any components you designate. The components are hosted in a process (package) called mtx.exe on NT 4.x and dllhost.exe on Windows 2000. These processes are standard processes that are managed through the MTC or CSMC.

To place the Java COM objects into the NameServer package (assuming you named the new package NameServer), double-click the package in the MTC or CSMC. Then, right-click the Components icon and click New then Component. In the CSMC, a dialog box will appear; just click Next. Then, in both the MTC and CSMC, click the Import Components button. Your computer should present you with a list that contains the progid of every COM component that can be hosted in an MTS package.

RMI does not inherit the level of complexity that surrounds RPC, CORBA, and DCOM

Search the list for the friendly names you assigned to the NameServer class and NameServerProxy class and highlight them. If you use the CSMC, click Next and then Finish. If you use the MTC, click Finish. At this point, both Java classes are hosted in the MTS package and you can access them like other COM components in the Microsoft operating system.

When you are finished, leave the NameServerProxy component properties as they are and change the NameServer component properties. To do this, right-click on the NameServer component and click the Properties option. A dialog box will appear that contains all available properties for the NameServer component (you can do many things to this component). Remember, the NameServerProxy component is only a proxy and should never contain business logic. You may want your NameServer component that contains all of the business logic to participate in or require a new transaction. Clicking on the Transactions tab in the Properties dialog box can set this property, or you can set security restraints on the COM object. If you use the CSMC, you can even set properties that pertain to the concurrency support for the component.

For fun, set the NameServer component to require a new transaction. This means that every RMI call to getNameImpl() will participate in a new transaction that starts when the getNameImpl() method marshals the RMI call to the NameServer Java COM component. You can also set other properties to use the capabilities of MTS.

Executing the Sample Code

Let's test the sample code and watch it marshal RMI calls over COM to Java classes hosted within MTS. You will notice that the main method in the NameServerProxy class can be run from the command line. This is important because the RMI server must be started and a registry (rmiregistry) must be instantiated for the client to make RMI calls to the proxy. To execute the NameServerProxy and start the registry, the NameServerProxy code must be executed using Microsoft's Jview tool. Jview is Microsoft's utility that executes Java applets and applications from the command line and is similar to the Java command-line utility that Sun provides in its JDK.

Execute the command-line procedure Jview NameServerProxy. Doing this will start the RMI server and prepare the proxy to accept RMI calls from clients.

Inside the MTC or CSMC, you can watch the state of the components change when they are called. Feel free to test the different views and learn what happens.

You can now execute the client code using a command-line utility—for example, java RMIClient. This helps validate that the RMI calls a client makes are properly marshaled into COM calls and into the NameServer component.

When the client is executed, it makes a standard RMI call to the RMI server hosted within MTS. The RMI server then propagates that call to the appropriate component hosted within MTS, and it returns a value as appropriate. The MTS component may participate in a transaction or a series of transactions before the return value is returned. Either way, the action is transparent to the client.

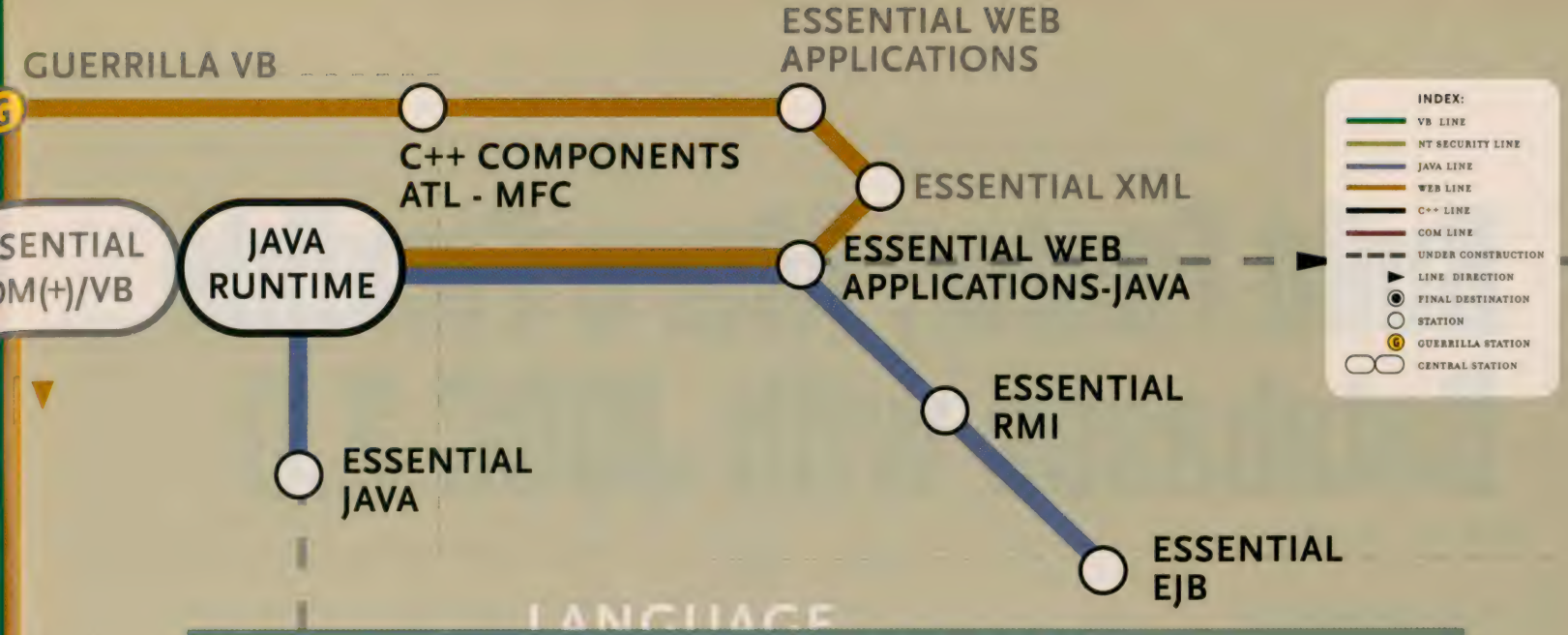
This solution may not be everyone's cup of tea, but MTS can be an inexpensive substitute for an Enterprise JavaBeans or cross-platform application server. By using this marshalling method, you can use MTS as a Java TPO. It meets many needs and has demonstrated the reliability and performance you expect from a TPO. Consider it when the time is right. ■

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0002 Download all of the code for this issue of *Java Pro*. Free online registration is required.
JP0002MD DevX Premier Club members download the code for this article.
JP0002MD_T DevX Premier Club members can read this feature online.

Not a member yet? Go to www.DevX.com to sign up for the free Registered Level or to subscribe to the Premier Club.



programmer's guide to

LEARNING JAVA

FOR DISTRIBUTED COMPUTING

FASTEST TRACKS

ESSENTIAL JAVA

This course is for professional developers currently working in another language who want a fast-paced and in-depth introduction into using Java so that they can create stand-alone applications immediately.

INTENSIVE JAVA

This course is about getting inside the Java virtual machine to better understand its expectations, logic, design and resulting behavior with your application. You'll learn many undocumented tricks and techniques that the VM uses and how you can surmount any optimization, deployment, and security problems in your Java application.

ESSENTIAL RMI

This course is about the technology behind building distributed systems withJava and the core RMI features in the JDK. You will be provided with a real understanding of how to get each feature to work for you and the strengths and weaknesses of each.

TRANSACTIONAL PROGRAMMING WITH EJB

This course will show you the programming model defined by the EJB's specification and builds on your knowledge of Java RMI.

ESSENTIAL WEB APPS-JAVA

This course focuses on writing distributed applications written in Java that use browser-based clients. You'll learn about Java applets and servlets and how to use Beans for managing data.

CITY CORE TIME TABLE

5-DAY HANDS-ON COURSES UNDERGROUND

SURVIVAL SKILL STOPS	TIME		LOCATION
	FEBRUARY	MARCH	
ESSENTIAL JAVA™ for developers		03/13-03/17	LOS ANGELES
	02/21-02/25		BOSTON
	02/28-03/03		PORTLAND
	02/07-02/11		ENGLAND
INTENSIVE JAVA™ the runtime environment		03/27-03/31	LOS ANGELES
		03/06-03/10	BOSTON
			PORTLAND
			ENGLAND
ESSENTIAL RMI Java distributed objects	02/07-02/11		LOS ANGELES
			BOSTON
			PORTLAND
			ENGLAND
TRANSACTIONAL PROGRAMMING WITH EJB		03/08-03/10	LOS ANGELES
		03/29-03/31	BOSTON
			PORTLAND
		03/01-03/03	ENGLAND
ESSENTIAL WEB APPS - JAVA for Java technologies	02/22-02/25		LOS ANGELES
			BOSTON
			PORTLAND
	02/01-02/04		ENGLAND

"tickets please"

TRAINING STATIONS

los angeles
boston
england
portland, or

REGISTRATION & INFORMATION

800.699.1932 *for the us*
08000.562.265 *within the uk*
+44 1242 525 108 *within europe*

DEVELOPMENTOR 2000

www.develop.com

Java is a registered trademark of Sun Microsystems.

Build Faster, More Flexible Databases with JDBC 2.0

The new connectivity standard improves both performance and functionality

by Brett Spell

What most people refer to simply as Java database connectivity, or "JDBC" is actually JDBC 1.2. Although JDBC 2.0 has been around as a specification for some time and is included with Java 2, most drivers didn't implement the new standard until recently. Now that such drivers are starting to appear, however, it's easier to implement the strengths of this new connectivity standard. In this, the first of three articles examining the many new features of JDBC 2.0, I'll examine the improvements JDBC 2.0 makes over the limitations of 1.2, and how the drivers implement them.

Let's start with a basic database scenario. For example, if your application needs to execute a series of Structured Query Language (SQL) statements to update information, you might code something like that shown in Listing 1.

Here, the first line creates an instance of a JDBC driver—specifically, the JDBC-ODBC bridge driver. The second statement obtains a connection to the database, and the next creates an instance of Statement and uses it to perform a series of updates. Note that each of the updates will be sent immediately to the database when executeUpdate() is called.

Similarly, if your application needs to read from the database, it could perform the same first three steps. However, instead of calling executeUpdate(), it might use executeQuery(), passing a SELECT statement as a parameter and getting back an instance of ResultSet. The result set maintains a pointer or "cursor" to one of the rows returned by the SELECT (initially pointing before the first row) and allows the user to move the cursor forward one row at a time by calling the next() method. For example, this code selects all rows from the INVENTORY table and prints out the string value stored in the second column of each row:

```
Statement stmt = conn.createStatement();
ResultSet rset = stmt.executeQuery(
  "SELECT * FROM INVENTORY");
while (rset.next()) {
  System.out.println("Item name " +
    rset.getString(2));
} // while (rset.next())
```

Performance Problems

One of the most common complaints about JDBC has been that it suffers from poor performance. Part of the responsibility for this problem lies with the interpreted nature of Java, but it's also related to the design of JDBC 1.2 and the drivers that implement it. For example, in the cases we just saw (which are typical of the way you would use JDBC), each one represents a situation where performance can be greatly improved:

- In the case of a SELECT, where we read rows from the database, the driver may request only a single row at a time from the database.
- Each INSERT, UPDATE, and DELETE statement issued through the executeUpdate() method is sent immediately to the database.

Listing 1 Needing SQL Statements If your application needs to execute a series of SQL statements to update information, you might code something like this.

```
new sun.jdbc.odbc.JdbcOdbcDriver();
Connection conn = DriverManager.getConnection(
  "jdbc:odbc:test", "bspell", "brett");
Statement stmt = conn.createStatement();
int result = stmt.executeUpdate(
  "UPDATE CUSTINFO SET STATUS = 1
  WHERE CUSTOMERNUM = 12345");
result = stmt.executeUpdate(
  "DELETE FROM ORDERS WHERE CUSTOMERNUM = 98765");
result = stmt.executeUpdate(
  "INSERT INTO INVENTORY VALUES (246, 'Chair')");
```

Brett Spell is a native of Westlake, Louisiana and has a degree in computer science from Louisiana Tech University. He lives in Plano, Texas, where he works for Pepsico Business Solutions Group at the Frito-Lay headquarters. Brett can be reached at javabrewer@aol.com.

These are really two variations of the same problem: each update and row retrieval requires a separate call to a database that typically resides on a different machine from the caller. In other words, each request generates a separate call across the network, and this is a problem because network communications usually involve a lot of overhead. Making many calls can be slow even with a very fast network, and database applications often do generate many requests. This issue isn't specific to database operations; it also becomes a factor in using distributed objects, such as with RMI, CORBA, or Enterprise JavaBeans (EJB). A simple and common solution to this problem is to group a set of requests and send them across the network in a single call. Even though the volume of data transmitted across the network will be about the same, this approach often improves an application's performance because network overhead is reduced.

At this point, you may be wondering how to group database requests. If you're using JDBC 1.2, the answer is that you can't, although the behavior specified for 1.2 doesn't prohibit the driver from pulling multiple rows across the network at one time. In fact, the documentation from JavaSoft states "We anticipate that drivers will normally prefetch rows in larger chunks so as to reduce the number of interactions with the target database."

This assumption, however, is problematic. For example, suppose your application performs a query that returns multiple rows, and the JDBC driver stores them in memory ("prefetches" them) until your application requests them. Let's also assume that another user modifies some of those rows in the database after the driver fetches them, but before your application reads them from the result set. Since the data prefetched by the driver is outdated, it will return information to your application that does not correctly reflect what is stored in the database. Depending on the transaction isolation level used, this may not be the expected behavior and can create problems for your application. (For more information on transaction isolation levels, see "Using Database Transactions In Java" in the September 1999 issue of *Java Pro.*) Still, many applications operate under the assumption that the data will not be modified while it's being read, or that such modifications aren't a problem. In those cases, it's acceptable for the JDBC driver to prefetch rows.

The problem with this is that when you're using JDBC 1.2, the application cannot specify to the JDBC driver that it should or should not allow prefetching. However, in JDBC 2.0, you *can* specify how many rows (if any) should be prefetched by the driver whenever more rows are needed. You can set this value by calling setFetchSize() for either a Statement or a ResultSet. Setting it for a statement makes it the default value for any result sets created by that statement, while setting it for a specific result set overrides the value "inherited" from its statement. For example, the following code sets a prefetch size of 10 for the statement, and that value is used by the first result set. The second result set then sets its own fetch size to 1, effectively disabling prefetching of rows:

```
Statement stmt = conn.createStatement();
stmt.setFetchSize(10);
ResultSet rset = stmt.executeQuery(
  "SELECT * FROM INVENTORY");
.
.
.
rset = stmt.executeQuery("SELECT * FROM CUSTINFO");
rset.setFetchSize(1);
```

One last point about the fetch size: this is simply a *suggestion* for the driver, which may or may not prefetch the number of rows you specify. Whether the driver does so depends entirely on the vendor's implementation. In some cases, prefetching may not be supported at all, but for drivers that do support this feature, setting the value appropriately can greatly improve the performance of your application. Even if you allow the driver to use prefetching, you still can ensure that you're not seeing outdated information, because JDBC 2.0 also provides a refreshRow() method that causes the result set to retrieve an updated copy of the current row from the database.

A driver can claim to be 2.0-compliant but still not support the features your application needs

Remember how each update statement such as INSERT, UPDATE, and DELETE is sent to and processed by the database immediately when the executeUpdate() method is used? If you've guessed that JDBC 2.0 also provides a way to send updates across the network in groups (or "batches"), you're correct. In fact, this feature is called "batch updating" and is very easy to use. Instead of using executeUpdate(), simply call addBatch(). Once you've added all the update statements that should be grouped together, use executeBatch() to send them to the database for execution. This code, which performs the same modifications as the code in Listing 1, illustrates this action:

```
conn.setAutoCommit(false);
Statement stmt = conn.createStatement();
stmt.addBatch("UPDATE CUSTINFO SET
  STATUS = 1 WHERE CUSTOMERNUM = 12345");
stmt.addBatch("DELETE FROM ORDERS WHERE
  CUSTOMERNUM = 98765");
stmt.addBatch("INSERT INTO INVENTORY
  VALUES(246, 'Chair')");
int[] results = stmt.executeBatch();
```

Note that in this example, we disabled auto commit mode before attempting the updates. You'll normally want to do this when using batch updates to ensure that either all of the updates are committed, or none of them are. If auto commit mode is left enabled and one or more statements fail, the successful changes will be committed anyway, which is not usually the desired behavior. Instead, you'll typically want to roll back all of the changes that were made if any of the update statements fail.

Improved Functionality

With JDBC 1.2, the data represented by a ResultSet is read-only and can be accessed only by moving forward through it one row at a time, beginning with the first row. However, JDBC 2.0 provides much greater flexibility, allowing you to set three properties related to how result sets function. Those properties are concurrency, scrolling, and sensitivity to changes made to the underlying data. (These last two properties are referred to collectively as the result set's "type.") These values are specified when you create a State-

ment instance, and all result sets created by the statement will "inherit" those characteristics. For example:

```
Statement stmt = conn.createStatement(
  resultSetType, resultSetConcurrency);
```

The parameter values should correspond to one of the integer constants defined in the ResultSet class: TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE, or TYPE_SCROLL_SENSITIVE for the result set type, and CONCUR_READ_ONLY or CONCUR_UPDATABLE for the concurrency value.

Concurrency mode refers to the ability to modify the data returned by a result set. If your application does not need to modify this data, specifying CONCUR_READ_ONLY for the concurrency mode parameter will cause the statement to create result sets that are read-only, just as in JDBC 1.2. On the other hand, specifying CONCUR_UPDATABLE allows your application to make changes to the data in the result set and have those changes stored in the underlying table(s) in the database.

You can insert new rows, delete rows, and change the values stored in the rows using new methods defined in ResultSet. For example, insertRow() inserts a new row at the current cursor position, while deleteRow() deletes the current row from the result set and the database. To change values stored in a row, JDBC 2.0 provides a new set of updateXXX() methods that modify the data in the result set's current row, just as the existing readXXX() methods allow you to retrieve the data. For each "read" method defined in JDBC 1.2, a corresponding "update" method is also defined in 2.0 (see Table 1.) In general, these changes are not reflected automatically in the database, except for deletions. However, after you have finished modifying an existing row or initializing a newly created one, you can call updateRow() to store your changes.

Data type	Read method (JDBC 1.2 & 2.0)	Update method (JDBC 2.0 only)
Boolean	boolean getBoolean(int col)	void updateBoolean(int col, boolean value)
Float	float getFloat(int col)	void updateFloat(int col, float value)
Int	int getInt(int col)	void updateInt(int col, int value)
Long	long getLong(int col)	void updateLong(int col, long value)
Object	Object getObject(int col)	void updateObject(int col, Object value)
String	String getString(int col)	void updateString(int col, String value)

Table 1 Reading and Writing with ResultSets This list shows some of the read/get methods available for obtaining row data and associated update methods introduced in JDBC 2.0 for use with updatable result sets. Each of these methods is overloaded and has a counterpart that allows you to specify the column's name instead of its index (not shown).

As with prefetching behavior, support for concurrency modes varies from one JDBC driver to the next. To ensure that your driver supports a given mode, use the supportsResultSetConcurrency() method in the DatabaseMetaData class. For more information on using metadata in JDBC, see "Enhancing Database Code with Metadata" in the June 1999 issue of *Java Pro*.

Result Set Types

A result set that is "scrollable" allows you to move the cursor to any row and both forward and backward through its data, instead of just moving forward as was the case in JDBC 1.2. The benefit of this strategy is clear when your application needs to display a huge amount of data. Organizations commonly have database tables with millions of records in them. If the application were to try to load all of the records into memory at once, the user's machine would very likely run out of memory before it could do so. A potential solution to this problem is to display only a portion of the table's records at a time and allow the user to move forward and backward through them. You've probably seen an example of this behavior already, because most Internet search engines use it when you perform a search that returns too many matches to display at one time.

You now have a variety of methods available for positioning a result set's cursor

However, this approach isn't feasible with JDBC 1.2, which allows you only to move forward through the result set. Once you've moved to (for example) record number one million, trying to get back to the previous record requires you to reissue the SELECT and call next() 999,999 times. Obviously, performance would suffer under this approach, and caching the rows already displayed would lead to the original problem of running out of memory. In JDBC 2.0, however, this "scrolling" design can be made to work, because in contrast to the simple next() method provided in JDBC 1.2, you now have a variety of methods available for positioning a result set's cursor (see Table 2).

Method	Description
Absolute(int row)	Positions the cursor at the specified row (the first row in the result set is 1, the second row 2, and so on). If a negative value is specified, then the cursor is positioned at an offset relative to the end of the result set instead of the beginning.
AfterLast()	Positions the cursor past the end of the result set (after the last row).
BeforeFirst()	Positions the cursor at the beginning of the result set (before the first row). This is the same location that the cursor is initially in when a result set is created.
First()	Positions the cursor at the first row.
Last()	Positions the cursor at the last row.
Next()	Moves the cursor forward one row. This is the only method available for cursor navigation with JDBC 1.2-style ("forward only") result sets.
Previous()	Moves the cursor to the previous row in the result set.
Relative(int rows)	Moves the cursor either forward or backward the specified number of rows. If the value specified is positive, the cursor is moved forward by that amount. Otherwise, it moves backward the indicated number of rows.

Table 2 Cursor Navigation New cursor navigation methods added in JDBC 2.0 are available only when working with scrollable result sets. For nonscrollable or "forward-only" result sets, using the next() method is the only way to reposition the cursor.

INFORMATION TECHNOLOGY IS DEVELOPING FASTER
THAN YOUR BRAIN. INFORM IT.
YOUR BRAIN IS HUNGRY. InformIT
Faster and faster in the new millennium. Faster and faster they say. When a single thought is all you've got, you're going to fade away. What do you think? Is there a link? A portal to the answers? What if there were such a site from which you might find content? Endless Information Technology content. Would you swim in the advice of experts? Take a dip in the knowledge pool? www.informit.com. Keep up. Stay current. Be cool.
©2000 informIT. All rights reserved. InformIT is a trademark of Macmillan USA.

Resources

The JDBC 1.2 Specification
http://java.sun.com/products/jdk/1.3/docs/guide/jdbc/spec/jdbc-spec.frame.html

The JDBC 2.0 Specification Update
http://java.sun.com/products/jdk/1.3/docs/guide/jdbc/spec2/jdbc2.0.frame.html

JDBC Drivers List
http://java.sun.com/products/jdbc/drivers.html

Finally, the third property that can be specified for result sets is their sensitivity to changes made to the data in the database while the result set is open. Those changes can be made either by an updateable result set itself (referred to as the result set's "own" changes) or by a different result set or statement (changes by "others"). The distinction between "own" and "others" is important, because JDBC 2.0 allows you to specify whether changes by "others" should be made visible to an open result set. JDBC 2.0 does not, however, allow you to specify whether a result set can see its "own" changes. This sensitivity to changes made by others and the scrollable property are specified together as one of these result set types:

- TYPE_FORWARD_ONLY: This represents a JDBC 1.2-type result set. Only forward movement through the rows in the result set is allowed, and changes made to the underlying data by "others" may or may not be visible, depending on the transaction isolation level and how the database handles the processing of the result set.
- TYPE_SCROLL_INSENSITIVE: Specifying this value indicates that the result set should be scrollable and that changes made to the data in the database by "others" while the result set is open should not be visible through that result set. In other words, the result set represents an unchanging ("static") view of its data and is insensitive to changes made to the underlying data in the database.
- TYPE_SCROLL_SENSITIVE: This specifies that the result set should be scrollable and that it may reflect (is sensitive to) changes made by "others" to the underlying data in the database after it has been opened. That is, the result set should display what is currently stored in the database, including changes made to the data after the result set was opened.

Not all drivers can support each of these types, and DatabaseMetaData provides a way to determine which ones are supported through its supportsResultSetType() method. However, even if a given result set type is supported, the driver may not fully support the expected behavior. For example, if you select a TYPE_SCROLL_SENSITIVE result set, the result set might be sensitive to inserts and deletions, but not updates. To determine what changes are detectable in a sensitive result set, use othersUpdatesAreVisible(), othersInsertsAreVisible(), and othersDeletesAreVisible().

The result set's type affects only sensitivity to changes made by others, and you cannot specify whether or not a modifiable result set is sensitive to its own changes. This behavior varies from one JDBC driver to the next, but the ownUpdatesAreVisible(), ownInsertsAreVisible(), and ownDeletesAreVisible() methods in DatabaseMetaData are provided so that you can determine what behavior to expect from the driver.

Enhanced Object Persistence

New Java-aware database management systems can create stored procedures written in Java (instead of proprietary vendor languages) and even enable you to create user-defined types (UDTs) that correspond to Java classes. You can then create tables with columns that allow you to store instances of the class associated with that column's UDT. With JDBC 2.0, you can obtain a list of the UDTs by using the getUDTs() method in DatabaseMetaData. This returns information on the types, including the name of each one and its corresponding class.

To store and retrieve instances of a class associated with a UDT, use the same writeObject() and readObject() methods that have been available since JDBC 1.2. For example, to write an instance of an Address object that has an associated UDT in the database, you might code something like this:

```
Address addr = new Address();
int custnum = 123;
.
.
.
PreparedStatement ps =
  conn.prepareStatement("UPDATE CUSTINFO
  SET CUSTADDR = ? WHERE CUSTNUM =
  " + custnum);
ps.setObject(1, addr);
```

Similarly, to read an instance of this class from the same row and column:

```
Address addr;
Statement stmt = conn.createStatement();
ResultSet rset = stmt.executeQuery(
   "SELECT * FROM CUSTINFO WHERE
  CUSTNUM = " + custnum);
  if (rset.next()) {
  addr = (Address)rset.getObject(
  "CUSTADDR");
}  //  if (rset.next())
```

When reading and writing object instances, they may be stored using Java's object serialization protocol, but the database vendor can choose to store the instances using some other proprietary persistence mechanism.

If you're convinced of the benefits of using JDBC 2.0 but don't know where to find a driver that supports it, Sun's Web site contains a comprehensive list of vendors, the types of drivers they provide, and the databases they support (see resource listing). Far more drivers still support 1.2, but that is changing quickly.

Incidentally, if it seems like you've read a lot of caveats here about drivers supporting particular features of 2.0, that's because you have. A driver can claim to be 2.0-compliant but still not support the features your application needs. However, over time, most drivers probably will support most or all of these features. In the meantime, make sure that the driver you plan to use supports the features you need by contacting the vendor or, if possible, testing it yourself.

JDBC 2.0 provides a simple way to send updates across the network

In the next installment of this series, we'll look at some of the new data types that are part of the SQL3 standard, and see how they're supported in JDBC 2.0. In the third and final article, we'll see what new features are available as standard extensions and how you can use them. Hopefully you've already seen some things that will be valuable to you in writing database applications, but believe it or not, this is just the beginning. ■

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0002 Download all of the code for this issue of *Java Pro*. Free online registration is required.

JP0002BS DevX Premier Club members download the code for this article.

JP0002BS_T DevX Premier Club members can read this feature online.

Not a member yet? Go to www.DevX.com to sign up for the free Registered Level or to subscribe to the Premier Club.

Accelerate your Java™ Graphics Development Today

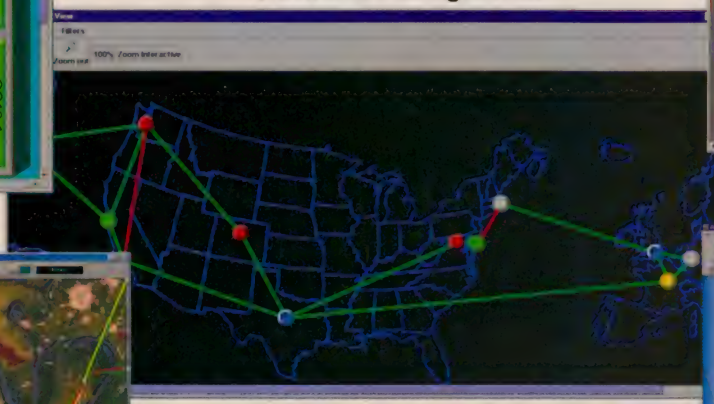
Ask about our free 30-day evaluation program

JLOOX

JLOOX is the best-in-class graphics toolkit for Java 2 developers. Based on Java2D, JLOOX provides a high-level 100% Java API that offers breakthrough performance and productivity gains for developers of graphics intensive Swing applications. It is ideal for applications requiring highly graphical displays such as networking and telecommunications,intelligent transportation systems, process control and simulation.

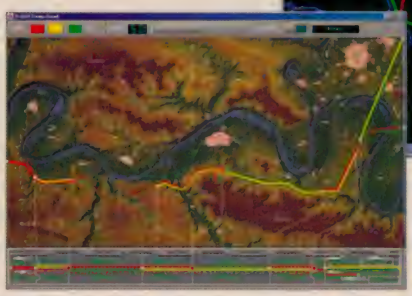
Element Management

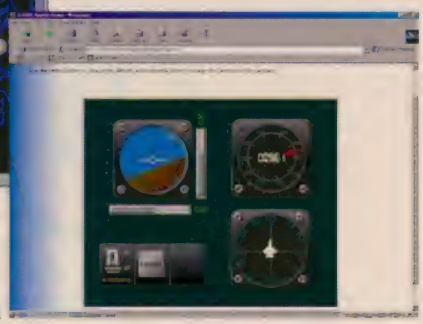
Network Management



SCADA

GIS





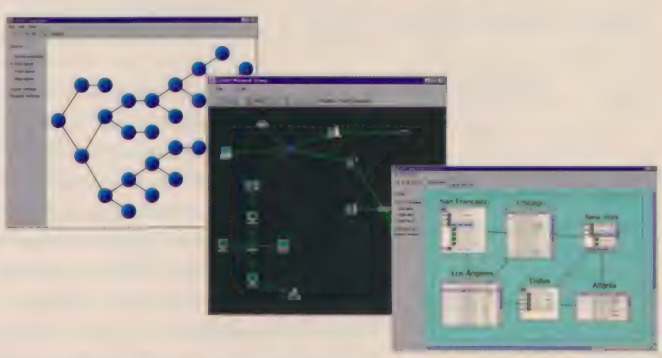
Instrumentation

Unique advantages:

- Includes an interactive editor for rapid creation and animation of graphic screens
- High-level intuitive API decreases development time
- Offers the rendering power of Java2D
- Pre-built collections of objects provide powerful data-visualization primitives
- Includes the award winning JClass Chart from KL Group
- Can be used with any IDE supporting Java 2
- Applications created with JLOOX have no runtime, royalty or deployment fees

JLOOX Layout

Easy visualization of complex topologies of JLOOX and Swing components



Accelerating Graphics Development™

LOOX Software Inc.

4962 El Camino Real, Ste. 206, Los Altos, CA 94022 • T:650-903-0942 F:650-903-9824 • www.loox.com

Copyright (c) LOOX Software Inc. 1999. LOOX is a registered trademark and JLOOX is a trademark of LOOX Software Inc. All other trademarks are the property of their respective owners.

Java to COM+: Making Cross-Platform Accessibility Work

Building reusable components that will be accessible from multiple platforms and environments isn't easy, but the right tools and some planning will help

by Bill Block and Jerry Brunning

Building a true enterprise system means connecting multiple operational departments *and* multiple computing platforms. Even applications designed to run standalone on one particular platform or operating system eventually may need to be accessible from, or provide services to, an otherwise incompatible platform. One department may use Java to deploy Web-based applications, while another department uses Visual Basic and COM+ to develop Windows applications. What happens when the Java team wants to use a piece of business functionality that the Visual Basic group developed?

To illustrate some of the issues facing developers building cross-platform accessible applications, we'll examine usage scenarios where cross-platform accessibility is required. Specifically, we'll show an in-depth example of one technique for getting a pure Java application to talk to a set of COM+ components running under Microsoft Component Service (formerly Microsoft Transaction Server).

Bill Block is a consultant with Clarity Consulting Inc., a Chicago-based firm specializing in the development of enterprise distributed systems. Bill manages Clarity's Distributed Computing Practice Area and has been involved with the design, development, and implementation of large client/server solutions. You can reach him at bblock@claritycon.com or visit Clarity's Web site at www.claritycon.com.
Jerry Brunning is also a consultant with Clarity Consulting. He wrote the white paper called "Building Year 2000–Compliant Applications with Visual Studio 6.0 and Microsoft Windows DNA" and speaks on Microsoft component technologies. You can reach him at jbrunning@claritycon.com.

Accessibility: Tougher than it Looks

Building applications that share information with each other is no easy task—especially when the applications in question were not built at the same time, or even by the same group in the company. Difficulties are almost guaranteed to arise when your application needs some services or functionality provided by an existing application.

Applications that were designed without the ability to share information with other applications must usually undergo a lot of rework. In some cases, the costs to retrofit an existing application can be significant, even approaching that of a rewrite.

The development tools you use can have a big effect on your ability to write multiplatform accessible applications, so you want to make the right decision. Just deciding on a standard set of development tools is a challenge for even a medium-sized IT department. While most corporate IT departments attempt to standardize on one tool or suite of tools for development, the fact is that many groups will use a variety of tools. Outside circumstances might require a particular tool, regardless of the corporate standard. Some factors that influence this decision include:

Application deployment options Is this a PC-based client/server application? Is it a Web-based transactional system? Is the application a set of COM+ middle-tier components that have no user interface at all?

The target OS platform Do you need to support clients running Windows 95? Do you require a specific service pack for your Windows NT clients?

The nature of the application Is a rich, capable user interface a requirement? Does it need to be multithreaded?

Legacy issues Do you need to interface with existing legacy systems?

Personnel requirements What is the expertise of the development staff? Do you need to hire outside contractors?

Production stage Is this a new application or the next revision of an application already in production?

With so many factors to consider, many developers find that limiting themselves to one set of programming tools is overly restrictive and sometimes even detrimental to their best efforts. Instead, development teams choose from a smorgasbord of tools, with broad, informal guidelines as to when to use a particular product. Too often the tool that

gets chosen for the project is the one that the developers have the most experience using, while other factors take a back seat. All this has the side effect of creating applications that don't communicate well.

What Developers Face

The upshot: Developers are left to sort out the complexities that drive businesses into multiplatform environments. Still, this is nothing new. We have always faced problems getting heterogeneous applications to talk to each other. Even in the best cases the scope of this problem is huge, from native messaging protocols to incompatible operating systems to desktop applications that need to exchange data. Even allowing your users the freedom to choose their own Web browser can have a major effect on your implementation plans. Should you abandon ActiveX controls on your intranet in favor of browser independence? Will your Java solution run under all of the major VMs? Throw into the mix backward compatibility, software distribution, and year 2000 compliance, and you can see how complex this issue becomes.

The complexity becomes more pronounced when you consider the design of a distributed architecture. With a significant piece of the development effort going toward building reusable business components shared by the enterprise, cross-platform accessibility becomes even more essential. Developers run into the inherent limitation of today's prevalent distributed architectures: they're somewhat proprietary. For example, DCOM works well with Microsoft technologies, but not with others, while CORBA and Enterprise Java Beans (EJB) are beginning to compete.

In an ideal environment, we would build systems with a keen understanding of today's business and tomorrow's direction, including an understanding of the need to exchange data with other applications. Sometimes, however, you get there too late. There still will be times when you need to link two or more applications that were built without anticipating this requirement. Developers have coped with this problem for years. So how have we adapted to this need that's been there all along, but seems to get only more complex? After all, applications are still built, deployed, and integrated, and—in some cases—still communicate with each other. We've continued to move forward, but sometimes past solutions have been mere workarounds, which we fell back on when we couldn't find a better solution, or because a better solution was too expensive.

In most cases, the easiest method for facing the need to have two applications communicate is to limit the functionality on the client—basically, to ignore the need. So, for example, when users employ multiple systems to perform different tasks, we may force them to log in to each system separately, instead of figuring out how to pass user credential and profile information from a legacy mainframe application to our client/server piece. This action presents only a minor nuisance to the user, but fixing it could be a big effort for a developer. However, dismissing the problem is not always so easy. In some situations, most of the company's business may depend on two enterprise applications being able to communicate.

IT departments often try to duplicate functionality in two applications rather than share it between them. For example, we may have a set of COM+ components enforcing our business rules. If we wanted to build a pure Java applet for one of our customers, the applet could not easily access our business components. Instead, we would have to duplicate the code in Java. Not only does this create a duplicate code base to maintain and deploy, but it also increases the chance that our order rules will not be applied correctly (maybe through a bug in the Java code or an oversight while translating the business rules). Duplicate functionality is a common real-world problem. In fact, companies often weigh the cost of reworking applications to share data against the cost of rewriting one or more of the applications.

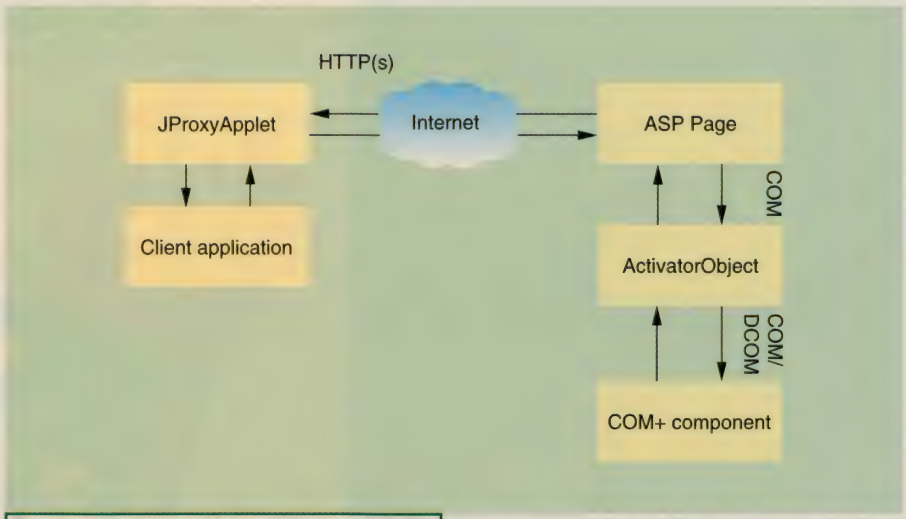


Figure 1 Anatomy of Cross-Platform COM+ An overview of the architecture. An application on Windows connects to a non-Windows platform by way of COM+-to-Java communication over the Internet. The JProxyApplet communicates with the COM component using XML over a secure HTTP connection.

Planning Better Solutions

Today, you have several options for performing distributed communication. First, DCOM rules the Microsoft world. DCOM is easy to use in client/server applications where both the client and server run a Microsoft operating system, but it is more difficult or impossible to use when trying to achieve interoperability between multiple environments. CORBA and Enterprise JavaBeans have been gaining support in non-Microsoft development groups, but, like DCOM, CORBA and EJB are often tied to proprietary hardware or software systems.

The cost of retrofitting an existing application can approach that of a rewrite

Another approach is IP messaging (for example, open a socket to port 9981 on server 127.0.0.1). This approach lets the client and server portions of your application communicate directly, but problems can still arise. For example, when an application needs to communicate through a proxy server or a firewall, it must now open ports on a firewall. Many system administrators frown on opening additional ports to the Internet. Even more problematic with the direct IP approach: Applications using pure socket connections are tied to a specific IP, so scalability quickly becomes an issue.

Of the many methods for communicating in a distributed environment, most are proprietary or require too much overhead to be cost-effective. Using these methods in a cross-platform environment, or even a cross-development envi-

Listing 1 Sending Messages This is an example of an XML message sent from a JProxyApplet to an ASP page.

```
  <methodcall>
    <progid>GeographyLibrary.PhoneBookClass</progid>
      <method>BrowseByNameAndAreaCode</method>
      <parameters>
        <parameter>
          <datatype>String</datatype>
          <value>John Smith</value>
      </parameter>
      <parameter>
          <datatype>Integer</datatype>
          <value>312</value>
        </parameter>
      </parameters>
      <returntype>Recordset</returntype>
</methodcall>
:

XML message returned from ASP page to JProxyApplet

  <methodreturn>
    <datatype>String</datatype>
    <value>312-555-1212</value>
<methodreturn>
```

TechTIP

Hex Your Bytes to Display

You may find yourself in a situation were you want to convert an array of bytes "byte []" into a hex string to display. The following method does exactly that.

```
String byteArrayToHexString(byte in[])
{
byte b = 0x00;
int i = 0;
if (in == null || in.length &lt;= 0)
return null;
String hexNums[] = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B",
  "C", "D", "E", "F"};
StringBuffer out = new StringBuffer(in.length * 2);

while (i &lt; in.length)
{
b = (byte) (in[i] & 0xF0); // Strip off high nibble
b = (byte) (b &gt;&gt;&gt; 4); // shift the bits down
b = (byte) (b & 0x0F); // must do this if high order bit is on!
out.append(hexNums[ (int) b]); // convert the nibble to a String Character
b = (byte) (in[i] & 0x0F); // Strip off low nibble
out.append(hexNums[ (int) b]); // convert the nibble to a String Character
i++;
}
return new String(out);
}
```

Say you have:

```
byte byteArray[] = {(byte)0xf1, (byte)0x0d, (byte)0x3c, (byte)0x44};
```

which is equivalent to "F10C3D22," you can display it as in:

```
String s = byteArrayToHexString(b);
System.out.println(byteArray)
```

Behrouz Fallahi

ronment, can get tricky. Engineering communication between DCOM and Java, for example, is messy—but it isn't impossible.

Integrating applications or platforms by purchasing a third-party tool to assist in development and deployment can work, but it often has high costs. Developers must learn how to integrate the new product into their applications, and system administrators must learn how to support new services. Moreover, third-party tools can limit the way applications communicate, because developers have little control over the means by which interoperability is achieved. Instead, that control is left up to the third-party tool designer.

What else can be done? While building your own cross-platform communication mechanism may sound like a huge task, it really isn't if you leverage existing systems and protocols.

Solving a Java Problem

We ran into a problem when one of our clients, who had been developing in 100% Pure Java, wanted to communicate with our COM+ components. Now we needed a way for a Java application to make what was essentially a DCOM call to our DLLs running under Microsoft Transaction Server.

The architecture we came up with used existing resources and knowledge to build a simple but comprehensive solution to our problem. The architecture consists of three main parts: a Java proxy/listener (for simplicity we'll refer to this piece as JProxyApplet), an ASP page hosted on our IIS server, and a business component used to "activate" our COM+ components (we'll refer to this object as the ActivatorObject). In a typical scenario (see Figure 1), the Java client sends a method request to the JProxyApplet component. The JProxyApplet next forwards that request to an ASP page, using query string parameters. The ASP page, in turn, calls the activator business component, which then calls the COM+ components and finally returns results in the response of the ASP page.

The most important step is creating a set of well-defined message layouts

The first, and arguably the most important, step in building this architecture is to create a set of well-defined message layouts. The messages sent between systems must be clear, yet must contain all the information necessary to process the requests. Extensible Markup Language (XML) offers a standard, well-supported

Order Your Favorite Tools and Save!

DEV·X com
The Development Exchange

Winter 1999/2000

MarketPlace

Tools for Programming Professionals

CONTENTS:

- Software for Download — Page 3
- Language Features — Page 4-5
- Internet Tools — Page 6
- Tools & Components — Page 7-9
- Microsoft® Featured Tools — Page 10-11
- Client/Server DBMS — Page 12-13
- Java Tools — Page 14-15
- Books & Certification — Page 16-17
- Productivity Tools — Page 18-19

DevX.com, Inc.
209 Hamilton Avenue
Palo Alto, CA 94301
Phone: 650-833-7100
Within the U.S. 1-800-848-5523

Microsoft® Visual Basic® 6.0
Save $60
See Page 11

$10 Instant Coupon
see page 2 for details

Dynamic HTML–20% Off
See Page 17

VideoSoft®
VSFlexGrid Pro 7
New Version
Download & Save!
See Page 9

Get Ready for the Next Wave of Application Development!

- **UPDATED SELECTION** of the best products for Web and Windows® 2000 development
- **EXPERT REVIEWS** from DevX.com — Read the DevX Review
- **AWARD-WINNING** products from the *VBPJ* and *Java Pro* Readers' Choice Awards
- **DOWNLOADABLE TOOLS** see listings and features on pages 3, 7-9 — Download Now
- **BONUS COUPON INSIDE** see page 2 for details

Premier Club Members!
Save an extra 10% OFF your software downloads! See details on page 19

Great Prices. Secure and Easy Online Ordering 24 Hours a Day!
http://software.devx.com • http://books.devx.com

WELCOME

Welcome to the DevX.com MarketPlace!

The DevX.com Marketplace features essential, award-winning productivity tools and components available either for immediate download purchase, or for shipment to your door. Our expanded bookstore includes titles to support the languages and tools you work with today. Plus we have easy ways to order our magazines and conferences online.

We cut through the clutter to bring you the tools that will help you, as an IT professional, get your job done better and faster.

- **40% off** weekly featured books
- **20% off** all books
- **up to 30% off** on all software
- **an additional 10%** all downloadable software for Premier Club Members
- **plus $10 savings** on all books and software over $40. ▼

$10 $10

For online use only

SAVE $10 INSTANTLY

on your next online software or book purchase*

your coupon code:

CM22X2

1. Go to the www.devx.com MarketPlace
2. Add your selected titles to your Shopping Cart
3. Enter your coupon code during check out

Offer expires March 15, 2000

*Get $10 off orders of $40 or more

$10 DEV·X MarketPlace $10

HTTP://SOFTWARE.DEVX.COM • HTTP://BOOKS.DEVX.COM

***Some restrictions apply:** MarketPlace coupons are for online redemption only and cannot be replaced if lost or stolen. The dollar value reflected on your coupon may be applied to your merchandise subtotal only, before shipping and taxes are charged. Returned items purchased with a coupon will be refunded the full amount you paid for each item. Coupons are not redeemable for cash and are good toward any book or software product purchased during your next visit on Devx MarketPlace. Conference registration, Magazine and CD subscriptions cannot be applied toward minimum. This offer cannot be combined with any other offers or promotions, other than our standard Premier Club discount offered on selected products.

Order Online at
http://software.devx.com
http://books.devx.com

Order and Download Your Favorite Tools Now!

Why wait? We carry the top development tools and components in downloadable format so you can start using them immediately to get your job done. Watch for our new inventory and price specials each week!

Download Now

Advantageware

- VB Advantage Version 6
  Pub: $299 • DevX Price: $259

Desaware, Inc.

- SpyWorks Professional Edition 6.1
  Pub: $379 • DevX Price: $319
- SpyWorks Standard Edition 6
  Pub: $199 • DevX Price: $169
- VersionStamper 6
  Pub: $249 • DevX Price: $219
- Gallimaufry 2.0
  Pub: $149 • DevX Price: $129
- StorageTools 2
  Pub: $149 • DevX Price: $129

FarPoint Technologies

- Button Objx
  Pub: $199 • DevX Price: $139
- Calendar Objx
  Pub: $159 • DevX Price: $119
- Daily PlanIt
  Pub: $299 • DevX Price: $219
- Input Pro 3.0
  Pub: $149 • DevX Price: $109
- List Pro
  Pub: $245 • DevX Price: $159
- Spread
  Pub: $349 • DevX Price: $249
- Tab Pro
  Pub: $129 • DevX Price: $99

Intara

- Foundation 3
  Pub: $295 • DevX Price: $279

KL Group

- JClass Chart Bytecode 4.0
  Pub: $399 • DevX Price: $349
- JClass LiveTable Bytecode 4.0
  Pub: $399 • DevX Price: $329
- JProbe Profiler 2.5,
  Developer Ed./Windows
  Pub: $499 • DevX Price: $429
- JProbe Profiler 2.5,
  ServerSide Ed./Solaris
  Pub: $649 • DevX Price: $589
- Jprobe Profiler 2.5,
  ServerSide Ed./Windows
  Pub: $649 • DevX Price: $589

ProtoView Development Corporation

- ActiveX Component Suite 7
  Pub: $695 • DevX Price: $619
- ActiveX Enterprise Suite 7
  Pub: $1495 • DevX Price: $1349
- Data Explorer 7
  Pub: $395 • DevX Price: $359

ProtoView Development Corporation

- DataTable 7
  Pub: $295 • DevX Price: $269
- JFCSuite 2.0
  Pub: $995 • DevX Price: $899
- JFCSuite Enterprise Edition 2.0
  Pub: $1,995 • DevX Price: $1,869
- JFCSuite Enterprise Edition
  (Source Code) 2.0
  Pub: $3,495 • DevX Price:$3,299
- JFCSuite With Source Code 2.0
  Pub: $2,495 • DevX Price: $2,199
- JSuite Enterprise Edition
  (Source Code) 2.5
  Pub: $2,995 • DevX Price: $2,799
- JSuite With Source Code 2.5
  Pub: $1,995 • DevX Price: $1,759
- Schedule X 6.5
  Pub: $395 • DevX Price: $359
- TreeViewX 6.5
  Pub: $199 • DevX Price: $179

Sheridan Software Systems

- ActiveListBar 1.01
  Pub: $139 • DevX Price: $99
- ActiveSuite Plus
  Pub: $595 • DevX Price: $495
- ActiveThreed Plus
  Pub: $179 • DevX Price: $115
- ActiveToolBars Plus
  Pub: $195 • DevX Price: $159
- ActiveTreeView 1.02
  Pub: $195 • DevX Price: $134
- Calendar Widgets 1.08
  Pub: $139 • DevX Price: $94
- Code Assist
  Pub: $295 • DevX Price: $219
- Components Suite
  Pub: $429 • DevX Price: $339
- Data Widgets 3.11
  Pub: $295 • DevX Price: $219
- Designer Widgets 2.02
  Pub: $139 • DevX Price: $95
- VBAssist 5.03
  Pub: $179 • DevX Price: $118

VideoSoft

- VS-OCX
  Pub: $149 • DevX Price: $129
- VSFlexGrid Pro
  Pub: $299 • DevX Price: $229
- VSVIEW6
  Pub: $299 • DevX Price: $259
- VSVIEW 6.0 Upgrade
  Pub: $159 • DevX Price: $139

Order and Download Online at

http://software.devx.com

Or call 800.848.5523 or 650.833.7100 (outside the USA)

http://software.devx.com

LANGUAGE FEATURES

TOOLS FOR PROGRAMMING PROFESSIONALS

From C++, to Visual Basic, Java, and More...

Microsoft® Visual Basic® 6.0

The world's most popular rapid application development tool!
It's estimated that 50% of all professional developers use award-winning Visual Basic. VB 6.0 ups the ante from previous versions with a number of powerful interface controls, integrated visual database tools, integrated reporting, and much more for creating your highest-performing enterprise and Web-based applications yet.

Also see page 11

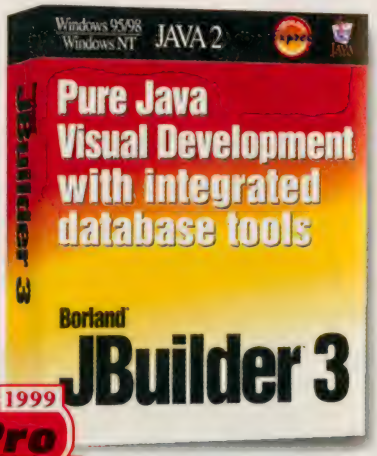
VERSION

- Professional Edition
  Format: CD ROM
  Pub: $549
  DevX Price: $489
- Enterprise Edition
  Format: CD ROM
  Pub: $1,299
  DevX Price: $1,199
- Professional Competitive Version Upgrade
  Format: CD ROM
  Pub: $279
  DevX Price: $249
- Enterprise Competitive Version Upgrade
  Format: CD ROM
  Pub: $759
  DevX Price: $689

Borland® JBuilder 3™

Create pure Java enterprise-scale applications for the Java 2 platform
JBuilder 3 delivers the key features required for productive Java development, including wizards and designers for creating reusable JavaBeans and Enterprise JavaBeans, visual tools and reusable components for rapidly creating platform-independent client/server applications, servlets, and applets.

Read the DevX Review



1999 JAVAPro READER'S CHOICE

VERSION

- Professional New User
  Format: CD ROM
  Pub: $799
  DevX Price: $559
- Professional Competitive Version Special Upgrade
  Format: CD ROM
  Pub: $299.95
  DevX Price: $289
- Enterprise New User
  Format: CD ROM
  Pub: $2,499
  DevX Price: $2,429
- Enterprise Upgrade to Professional
  Format: CD ROM
  Pub: $2,199
  DevX Price: $2,109
- Client/Server Enterprise Upgrade for Windows 95, 98, and NT
  Format: CD ROM
  Pub: $1,699
  DevX Price: $1,599

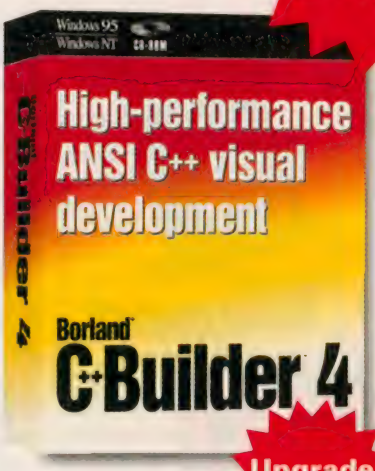
Order Online at
http://software.devx.com

Go to DevX.com for a complete list of our C++ products

Borland® C++ Builder™ 4 ▶

Deliver applications to market faster with C++ Builder 4!
This integrated development environment combines a 32-bit C++ compiler and debugger, including a visual component library and dual support for COM and CORBA. C++Builder 4 is an ANSI/ISO C++ compiler and visual development environment for rapidly creating powerful high-performance Windows® and distributed applications that are interoperable across multiple platforms. It has the fastest and highest-quality Windows C++ user interface development, and the most powerful and productive MTS, COM, and ActiveX support available for C++.

Read the DevX Review



Upgrade Today!

VERSION ▲

- Standard Edition
  Format: CD ROM
  Pub: $99.95
  DevX Price: $89
- Professional Edition New User
  Format: CD ROM
  Pub: $799.95
  DevX Price: $549
- Professional Competitive Upgrade
  Format: CD ROM
  Pub: $299.95
  DevX Price: $289.95
- Enterprise Edition New User
  Format: CD ROM
  Pub: $2,499
  DevX Price: $2,399
- Enterprise Edition Upgrade
  Format: CD ROM
  Pub Price: $1,699
  DevX Price: $1,599
- Enterprise Upgrade from Professional Edition
  Format: CD ROM
  Pub: $2,199
  DevX Price: $2,099

Microsoft® Visual C++™ 6.0 ▶

Take your productivity to a new level without sacrificing flexibility, performance, and control.
In addition to features like IntelliSense Technology, and Edit and Continue, which significantly speed development time, Visual C++ 6.0 contains greatly improved support for Web and Enterprise development.

VERSION ▲

- Standard Edition
  Format: CD ROM
  Pub: $109
  DevX Price: $99
- Professional Edition
  Format: CD ROM
  Pub: $549
  DevX Price: $479
- Professional Competitive Version Upgrade
  Format: CD ROM
  Pub: $279
  DevX Price: $249
- Enterprise Version Upgrade
  Format: CD ROM
  Pub: $759
  DevX Price: $699
- Enterprise
  Format: CD ROM
  Pub: $1,299
  DevX Price: $1,199

Save $100 on Enterprise Edition

SAVE $10 on your next online purchase
(see page 2 for details)

http://software.devx.com

Order Online at
http://software.devx.com
Or call 800.848.5523 or 650.833.7100 (outside the USA)

INTERNET TOOLS

Essential Tools For Web Development

Microsoft® Front Page™ 2000

Create and manage great-looking Web sites easily

The Microsoft FrontPage 2000 Web site creation and management tool gives you everything you need to create and manage exactly the site you want, whether you're creating a personal Web page or a corporate Internet or intranet site. And because it works like the rest of Microsoft Office Premium, FrontPage 2000 is easy to learn and use.

Read the DevX Review

VERSION:

- Windows 9X/NT
  Format: CD ROM
  Pub: $149
  DevX Price: $135
- Windows 9X/NT
  Version Upgrade
  Format: CD ROM
  Pub: $59.95
  DevX Price: $54

Pervasive Tango™ 2000

A powerful solution for your Web-based applications

Tango gives you all the power, flexibility, security and ease-of-use you need to deliver today's complex business-critical web solutions. Tango features a superior development environment, a highly scalable application server, and a host of other features—including Pervasive.SQL in all Microsoft Windows versions—that dramatically lower the total cost of ownership.

VERSION:

- Application Server
  Standard Edition
  for Windows NT
  Format: CD ROM
  Pub: $3,495
  DevX Price: $2,599
- Application Server
  Small Business Edition
  for Windows NT
  Format: CD ROM
  Pub: $1295
  DevX Price: $999
- Application Server
  Professional Edition
  for Windows NT
  Format: CD ROM
  Pub: $10,000
  DevX Price: $8,999

Read the DevX Review!

For even more information and expert reviews on many of the software products featured in this brochure visit the Product Guide at www.devx.com.

Order Online at
http://software.devx.com

DevX.com
Premier Club Members
Save an extra 10%
(See page 19 for details)

Customize Your Applications

Desaware VersionStamper 6.0

Download Now

Easily detect component version incompatibilities on your end-user's systems.

Read the DevX Review

VersionStamper provides a complete distribution solution. Determine the component dependencies of any project, detect conflicts based on embedded or remote file lists, and get automatic notification or software updates. VersionStamper 6 adds new lightweight ATL based components, controls and emergency rescue program. A new dependency wizard determines the components used by an application and creates scripts for verification and remote updating.

VERSION:
■ Format:
Download Version
Pub: $249
DevX Price: $219

New Version!

Download Now

FarPoint™ Input Pro 3

Easily create professional data-entry screens using Input Pro's eight custom controls.

Take advantage of the six formatted edit controls to quickly validate data and alert your user as they enter information or as the application programmatically assigns or edits values. Use the boolean control to display a check box, option button, or custom true/false/grayed value appearance. Use the memo control to display large amounts of text, even those greater than 64K.

VERSION:
■ Format:
Download Version
Pub: $149
DevX Price: $109

Download Today!

Download Now

ProtoView ActiveX Suite 7.0

ActiveX Suite bundles four of the best selling ActiveX products on the market into one low-priced toolkit

The ActiveX Component Suite has full support for the entire line of Microsoft Visual Studio 6 products (including Internet Explorer) and contains Microsoft Explorer-style UI components, grid, drop-in replacement ListBox and ComboBox, data input and validation components, treeview and scheduling tools.

VERSION:

■ Component Suite
Format:
Download Version
Pub: $695
DevX Price: $619

■ Enterprise Suite
Format:
Download Version
Pub: $1495
DevX Price: $1299

Order Online at
http://software.devx.com
Or call 800.848.5523 or 650.833.7100 (outside the USA)

TOOLS & COMPONENTS

Seagate Info™ 7 Cross Platform

Manage and distribute information efficiently across your company

Built atop a scalable, multi-tier architecture, Seagate Info gives users the ability to easily schedule reports, queries and multi-dimensional cubes and distribute them to the right people all across the organization. By providing the same accurate information to all those who need it, as well as the ability to drill into it with integrated ad hoc query and multi-dimensional OLAP technology, companies can more effectively address the competitive opportunities and challenges they face every day.

Read the DevX Review

Visual Basic Programmer's Journal '99 Award Readers' Choice Merit

VERSION

■ Starter Pack
Format: CD ROM
Pub: $1,870
DevX Price: $1,819

■ Starter Pack w/subscription
Format: CD ROM
Pub: $2,340
DevX Price: $2,299

■ Starter Pack Upgrade
Format: CD ROM
Pub: $975.00
DevX Price: $959

■ Starter Pack Upgrade with Subscription
Format: CD ROM
Pub: $1,440
DevX Price: $1,399

■ 5U Client License with Subscription
Format: CD ROM
Pub: $1,845
DevX Price: $1,799

■ 5U Client License Upgrade with Subscription
Format: CD ROM
Pub: $1,135
DevX Price: $1,099

■ 5U Client License Upgrade
Format: CD ROM
Pub: $765
DevX Price: $749

■ 5U Designer Module
Format: CD ROM
Pub: $1,995
DevX Price: $1,929

■ 5U Designer Module with Subscription
Format: CD ROM
Pub: $2,470
DevX Price: $2,399

■ 5U Designer Module Upgrade
Format: CD ROM
Pub: $1,025
DevX Price: $999

■ 5U Designer Module Upgrade with Subscription
Format: CD ROM
Pub: $1,520
DevX Price: $1,489

Sheridan™ Component Suite 2.0

Download Now

Three popular Sheridan components in one value-priced package!

Calendar Widgets—the perfect set of Sheridan Reusable Components for incorporating date-and time-related information in your applications. Besides being a great fit for a typical Personal Information Manager (PIM), Calendar Widgets is a natural for data entry, accounting, billing, project management, or any other application which uses time and dates.

Read the DevX Review

Data Widgets 3.0— set of six ActiveX controls for designing front-ends for your database applications. The enhanced new features of Data Widgets' 3.0 DataGrid gives you the freedom to print to paper or export to HTML. You can also export data to ASCII-delimited text files.

Designer Widgets 2.0.—Sheridan Reusable Components designed to give your application interfaces state-of-the-art look and feel. Designer Widgets 2.0 is loaded with the most enhanced interface controls, including Dockable Toolbar, the Notebook Tab Control, the Index Tab Control, and FormFX.

Visual Basic Programmer's Journal '99 Award Readers' Choice

VERSION
Format: Download Version
Pub: $429
DevX Price: $319

Order Online at
http://software.devx.com

VideoSoft® FlexGrid Pro 7 ▶

Customizing the display and presentation of dynamic data for Windows, Internet or intranet applications is now faster and easier than ever.

Read the DevX Review

VideoSoft's VSFlexGrid Pro 7.0 offers a rich feature set for quickly building flexible, powerful front ends – without adding user overhead. VSFlexGrid Pro 7.0 is 100% backward compatible with the MSFlexGrid in Visual Basic, so upgrading an existing application is easy. It supports a host of new data formatting options like freezing rows and columns, binding grid columns to image lists and a wallpaper property to enhance your application's appearance. And a new "light" version minimizes footprints for apps that don't require database support.

Download Now

New Version!

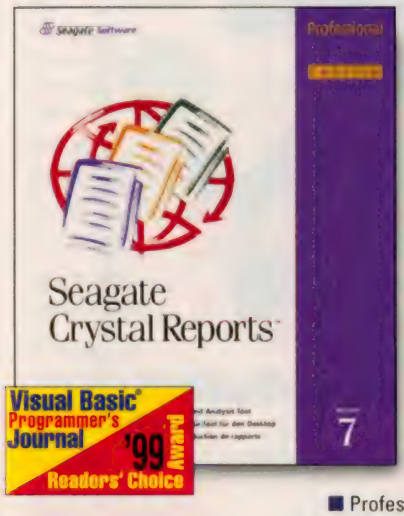
Visual Basic Programmer's Journal '99 Award Readers' Choice

VERSION ▲

■ Format: Download Version

Pub: $299

DevX Price: $229



◀ Seagate Crystal Reports™ 7

Featuring new VB6 ActiveX designer!

Provides developers, IT professionals and power users with a complete range of presentation-quality reporting features including advanced reporting over the Web and powerful tools for integrating reports into applications. Build virtually any report you can imagine: subreports, conditional reports, summary reports, cross-tabs, form reports, multiple details, mailing labels and more.

Read the DevX Review

VERSION ▲

■ Standard Edition
Format: CD ROM
Pub: $149
DevX Price: $119

■ Standard Edition Upgrade
Format: CD ROM
Pub: $79
DevX Price: $75

■ Standard 5U
Format: CD ROM
Pub: $495
DevX Price: $449

■ Standard 5U Upgrade
Format: CD ROM
Pub: $395
DevX Price: $359

■ Professional Edition
Format: CD ROM
Pub: $395
DevX Price: $289

■ Professional Edition Upgrade
Format: CD ROM
Pub: $199
DevX Price: $189

■ Professional Edition w/Subscription
Format: CD ROM
Pub: $499
DevX Price: $449

■ Professional Edition Upgrade w/Subscription
Format: CD ROM
Pub: $299
DevX Price: $289

■ Professional Edition 5U
Format: CD ROM
Pub: $1,495
DevX Price: $1,149

■ Professional Edition 5U Upgrade
Format: CD ROM
Pub: $859
DevX Price: $809

■ Professional Edition 5U w/subscription
Format: CD ROM
Pub: $1,950
DevX Price: $1,799

■ Professional Edition 5U Upgrade and w/subscription
Format: CD ROM
Pub: $1,359
DevX Price: $1,299

Order Online at
http://software.devx.com
Or call 800.848.5523 or 650.833.7100 (outside the USA)

http://software.devx.com

MICROSOFT FEATURES

Featured Products To Get You There Faster

Microsoft® Visual Studio™ 6.0

Develop for the next generation of Enterprise Web applications with Visual Studio 6.0

Includes the latest new versions of the Microsoft Visual Tools, enabling you to build scalable applications for Windows® and the Web. Suite Includes: Visual Basic®, Visual C++®, Visual FoxPro®, Visual InterDev™, and Visual J++®. Easy-to-create and reuse components easily with all of the Visual Studio tools; improved integration and productivity enhancements across all the Visual Tools.

VERSION

- Professional
  Format: CD ROM
  Pub: $1,079
  DevX Price: $1,029
- Professional Competitive Version Upgrade Format: CD ROM
  Pub: $549
  DevX Price: $519
- Enterprise Edition
  Format: CD ROM
  Pub: $1,619
  DevX Price: $1,499
- Enterprise Competitive Version Upgrade
  Format: CD ROM
  Pub: $979
  DevX Price: $949
- Enterprise Version Upgrade
  Format: CD ROM
  Pub: $1,079
  DevX Price: $1,039

Visual Basic® Programmer's Journal '99 Award Readers' Choice

Microsoft® VisualSourceSafe™ 6.0

Store and revise your Web content, and manage software development projects

Get secure and scalable version control, project-oriented version control for Web and PC content management, plus powerful, yet easy-to-use tools.

The Microsoft Visual SourceSafe 6.0 version control system is the latest edition of Microsoft's award-winning version control system for managing software and Web site development. Fully integrated with the Visual Basic, Visual C++, Visual J++, Visual InterDev, and Visual FoxPro development environments, as well as with Microsoft Office applications, it provides easy-to-use, project-oriented version control.

Read the DevX Review

VERSION

- Windows 95, Windows 98, and Macintosh
  Format: CD ROM
  Pub: $549
  DevX Price: $489
- Windows 98, and Windows NT, Competitive Version Upgrade
  Format: CD ROM
  Pub: $279
  DevX Price: $269

Get detailed information about every book and software product you see on DevX.com MarketPlace

Order Online at
http://software.devx.com

Microsoft® Visual Basic® 6.0

Deliver high-performance enterprise and Web-based applications—fast!
Get ready for Windows 2000, with the most popular tool for building data-centric business applications. Use the Visual Basic 6.0 integrated Visual Database Tools and new Data Environment Designer to design Oracle and Microsoft SQL Server databases and create reusable data access queries—all without leaving the Visual Basic environment. Build server-side Web applications that are easily accessible from any browser on any platform with Visual Basic 6.0 WebClasses.

Also see page 4

VERSION

- Professional
  Format: CD ROM
  Pub: $549
  DevX Price: $489
- Professional
  Competitive Version Upgrade
  Format: CD ROM
  Pub: $279
  DevX Price: $249
- Enterprise Edition
  Format: CD ROM
  Pub: $1,299
  DevX Price: $1,199
- Enterprise
  Competitive Version Upgrade
  Format: CD ROM
  Pub: $759
  DevX Price: $689

Includes Windows 2000 Readiness Kit

Microsoft® Visual J++ 6.0

Harness the productivity of the Java language, and the power of Windows, to build and deploy high-performance client/server solutions for Windows and the Web.

VERSION

- Standard Edition
  Format: CD ROM
  Pub: $109
  DevX Price: $99
- Professional Edition
  Format: CD ROM
  Pub: $549
  DevX Price: $499
- Professional
  Competitive Version Upgrade
  Format: CD ROM
  Pub: $219
  DevX Price: $209

Read the DevX Review

Microsoft

Microsoft® Visual Inter Dev™ 6.0

Design, build, debug, and deploy cross-platform Web applications faster than ever before
Visual InterDev 6.0 also features a new integrated WYSIWYG editor for ASP and Dynamic HTML pages, enhanced database programming tools, and end-to-end debugging facilities for multi-tier applications built with HTML and Script.

VERSION

- Professional
  Format: CD ROM
  Pub: $549
  DevX Price: $499
- Professional
  Competitive Version Upgrade
  Format: CD ROM
  Pub: $279
  DevX Price: $249

Order Online at
http://software.devx.com
Or call 800.848.5523 or 650.833.7100 (outside the USA)

http://software.devx.com

CLIENT/SERVER DBMS

Connect People with Information

IBM® DB2™ Universal 6.1

Powerhouse database for your e-business solutions

DB2 Universal 6.1 offers an open, industrial-strength database management tool for all your business intelligence, transaction processing, and business applications. Includes enhanced functionality and new features for integrating Web-based applications with your existing business apps, while maintaining the capacity you need.

VERSION

- Database Workgroup Edition Single User for Cross Platforms
  Format: CD ROM
  Pub: $249
  DevX Price: $229
- Developer's Edition for Cross Platforms
  Format: CD ROM
  Pub: $999
  DevX Price: $839
- Personal Edition for Cross Platforms
  Format: CD ROM
  Pub: $369
  DevX Price: $339
- Workgroup Edition for 1 Server/1 User for Cross Platforms
  Format: CD ROM
  Pub: $999
  DevX Price: $819

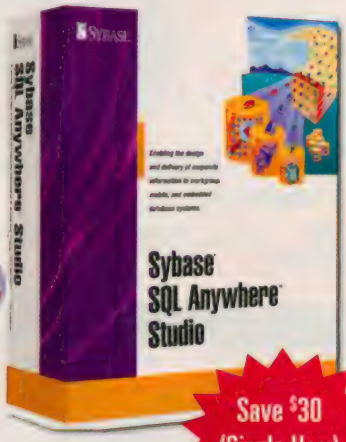
More detailed information about every book and software product you see can be found on the DevX.com MarketPlace

Sybase® SQL Anywhere™ Studio 6

Enables the design and delivery of corporate information to workgroup, mobile, and embedded database systems

Featuring Adaptive Server Anywhere™, Sybase's industry-leading mobile database, SQL Anywhere Studio delivers rich database functionality, and powerful replication optimized for occasionally connected environments, and a full suite of development and productivity tools.

Read the DevX Review



Save $30 (Single User)

VERSION

- Single User Version
  Format: CD ROM
  Pub: $399
  DevX Price: $369
- 5U Version
  Format: CD ROM
  Pub: $999
  DevX Price: $859
- Additional Network Seat License
  Format: License
  Pub: $645
  DevX Price: $599
- 5 Pack Additional Concurrent License
  Format: License
  Pub: $645
  DevX Price: $599

Also by Sybase....

Adaptive Server
Format: CD ROM

PowerBuilder
Format: CD ROM

PowerDesigner
Format: CD ROM

Order Online at
http://software.devx.com

Microsoft® SQL Server™ 7.0 ▶

Upgrade to the most robust database for the Windows family yet

Take your company to the next level with this powerful upgrade to Microsoft's BackOffice RDBMS. Simplified setup and administration, automated self-tuning, large database capabilities, row-level update and insert locking, improved replication and backup/restore, and native OLE DB support. A highly scalable solution, SQL Server integrates seamlessly with Windows NT™ Server and the rest of the Microsoft BackOffice™ family.

Read the DevX Review

Improved Optimization!

VERSION ▲

- 5 Users for WIN NT
  Format: CD ROM
  Pub:$1399
  DevX Price: $1259
- 10 Users for WIN NT
  Format: CD ROM
  Pub: $1999
  DevX Price: $1809
- 5 Users Competitive Version Upgrade for WIN NT
  Format: CD ROM
  Pub: $699
  DevX Price: $649

Merant™ DataDirect® 3.x ▶

Data Direct helps organizations become more agile by providing flexible data access and integration across new and existing data assets and systems

A comprehensive set of cross-platform ODBC drivers that leverage existing DBMS vendor middleware to deliver high-performance, reliable connectivity to all major databases. DataDirect Connect ODBC are the only drivers to consistently support the latest ODBC specification, databases and operating systems, providing the most advanced functionality available. Optimized to perform in high-volume environments.

MERANT

VERSION ▲

- ODBC Driver
  Format: CD ROM
  Pub: $99
  DevX Price: $89
- ODBC Pack
  Format: CD ROM
  Pub: $499
  DevX Price: $449

Also by Merant....

Merant Connect OLE DB Pack v1.0
Format: CD ROM
Pub: $499
DevX Price: $479

DEVX

Get news, tips, and article updates

- ✓ Weekly DevX Hotlinks News
- ✓ VB Update
- ✓ Java Update
- ✓ Enterprise Update
- ✓ ASP Update
- ✓ SQL Update
- ✓ Web Development
- ✓ C++ Update
- ✓ XML Update
- ✓ Exchange Update
- ✓ MarketPlace Specials

email address Get it!

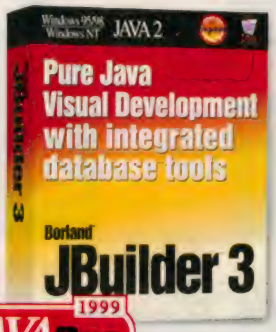
http://software.devx.com

Order Online at
http://software.devx.com
Or call 800.848.5523 or 650.833.7100 (outside the USA)

JAVA TOOLS

TOOLS FOR PROGRAMMING PROFESSIONALS

Comprehensive Resources to Build Enterprise-Scale Applications



JAVAPro 1999 READER'S CHOICE

Borland® JBuilder™ 3

Create pure Java enterprise-scale applications for the Java 2 platform

JBuilder 3 delivers a comprehensive set of tools for productive Java development, including numerous Wizards and designers for creating reusable JavaBeans and Enterprise JavaBeans, Visual tools and reusable components for rapidly creating platform-independent client/server applications, servlets, and applets.

Read the DevX Review

VERSION

- Standard Edition New User Version Format: CD ROM
  Pub: $99.95
  DevX Price: $91
- Professional Edition New User Version Format: CD ROM
  Pub: $799
  DevX Price: $559
- Professional Edition Competitive Version Special Upgrade Format: CD ROM
  Pub: $299.95
  DevX Price: $289
- Enterprise Edition Upgrade to to Professional Format: CD ROM
  Pub: $2,199
  DevX Price: $2,109
- Enterprise Edition New User Format: CD ROM
  Pub: $2,499
  DevX Price: $2,429
- Client/Server Enterprise Edition Upgrade Format: CD ROM
  Pub: $1,699
  DevX Price: $1029

IBM® VisualAge for Java™ 2.0

Build Web-enabled enterprise applications with IBM's VisualAge for Java. VisualAge leads the industry with proven support for building and testing Java applets, servlets, and Enterprise JavaBean components. It's the only Java development environment that supports the development and management of Java programs that can scale from Windows NT to OS/390 application servers.

Read the DevX Review

VERSION

- Professional Edition Upgrade - Single user Format: CD ROM
  Pub: $29
  DevX Price: $29
- Enterprise Edition for AIX Format: CD ROM
  Pub: $2,745
  DevX Price: $2,589
- Enterprise Edition Format: CD ROM
  Pub: $2,745
  DevX Price: $2,589
- Professional Edition Format: CD ROM
  Pub: $99
  DevX Price: $81
- Professional Edition Upgrade Format: CD ROM
  Pub: $29
  DevX Price: $25
- **Also by IBM... Visual Age for C++**

Order Online at
http://software.devx.com

Extra Savings! Use your $10 coupon and save more! (see page 2 for details)

Symantec™ VisualCafé™ 3.0

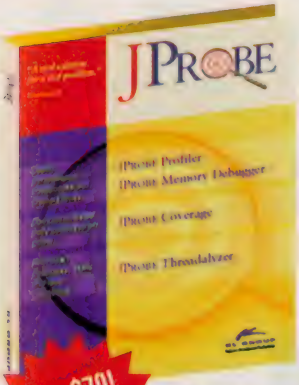
An easy to use RAD Java IDE featuring an integrated editor, debugger and compiler
VisualCafé delivers an extensive array of wizards, utilities and a professional library of 100+ JavaBeans with source code to help jump-start your Java development. VisualCafé's two-way drag and drop programming ensures that the code editor and visual designer are always synchronized for maximum productivity, letting you produce source code in real-time.

1999 JAVAPro READER'S CHOICE

VERSION

- Database Edition
  Format: CD ROM
  Pub: $799
  DevX Price: $789
- Professional Edition
  Format: CD ROM
  Pub: $299
  DevX Price: $279
- Standard Edition
  Format: CD ROM
  Pub: $99.95
  DevX Price: $89

Download Now



Save $70! (Developer Edition)

KL Group™ JProbe Profiler™ 2.5

A fully integrated set of tools to debug thread, memory and performance problems in Java applications
Quickly eliminate performance bottlenecks caused by inefficient objects and algorithms in your Java code. JProbe Profiler combines a visual call graph interface and unique data collection technology to provide you with highly accurate diagnostics on both performance and memory usage.

VERSION

- Developer Edition for Windows
  Format: Download Version
  Pub: $499
  DevX Price: $429
- ServerSide Edition for Solaris
  Format: Download Version
  Pub: $649
  DevX Price: $589
- ServerSide Edition for Windows
  Format: Download Version
  Pub: $649
  DevX Price: $589

Download Now

ProtoView JFCSuite 2.0

Lightweight components with powerful features JFCSuite is a component set that will "simplify and streamline" the development of graphical user-interface (GUI) components. JFC provides the basis for creating "lightweight" components that are smaller in size and don't rely on operating system-centric code to run. JFC also creates a look and feel that allows components to always look and operate like the platform they are set to run on. Includes JFCDataTable, JFCDataTree, JFCDataCalendar, JFCDataExplorer, and JFCDataInput.

VERSION

- JFCSuite Enterprise Edition Download Version
  Pub: $1,995
  DevX Price: $1,869
- JFCSuite Enterprise Edition with Source Code Download Version
  Pub: $3,495
  DevX Price: $3,299
- JFCSuite Download Version
  Pub: $995
  DevX Price: $899
- JFCSuite with Source Code Download Version
  Pub: $2,495
  DevX Price: $2,199

Order Online at
http://software.devx.com
Or call 800.848.5523 or 650.833.7100 (outside the USA)

http://software.devx.com

BOOKS/CERTIFICATION

DevX.com carries books that will help you do your job better and faster.
Here's just a glimpse of the topics in our inventory:

Deploying Microsoft Office 2000 Notes from the Field
Provides detailed real-world deployment information, including guidelines, implementation information, and recommended tools.
Paperback w/CD *By Microsoft Press*
Pub: $39.99
DevX Price: $31.99

Dynamic HTML
The HTML Developer's Guide—a valuable resource to help you master this important new skill. Explains principles and uses a learn-by-approach example.
Paperback w/CD *By Jeff Rule*
Pub: $39.95
DevX Price: $31.96

Dan Appleman's Win32 API Puzzle Book and Tutorials for Visual Basic Programmers
Turbocharge VB by understanding and controling the entire Windows API.
Paperback w/CD *By Dan Appleman*
Pub: $39.95
DevX Price: $31.96

SAMS Teach Yourself HomeSite 4 in 24 Hours
In just 24 sessions, you'll be up and running. Uses a straightforward, step-by-step approach.
Paperback w/CD
By Ben Forta
Pub: $24.99
DevX Price: $19.99

Developing Applications using Outlook 2000, CDO, Exchange, and Visual Basic
Provides a comprehensive reference to working with Microsoft's powerful collaborative development environment.
Paperback *By Raffaele Piemonte & Scott Jamison*
Pub: $54.95
DevX Price: $43.96

Visual Basic Controls in a Nutshell
Depth and quality on VB controls. A great addition to your library.
Paperback
By Evan Dictor
Pub: $24.95
DevX Price: $19.96

Hitchhiker's Guide to VB & SQL Server, Sixth Edition
The definitive book for developers who want to use VB to access Microsoft SQL Server.
Paperback w/CD *By William R. Vaughn*
Pub: $49.99
DevX Price: $39.99

Sams Teach Yourself Java 2 Online in Web Time
Want time to advance your skills through book and online instruction?
Paperback w/CD
By Stephen Gilbert, Bill McCarty
Pub: $49.99
DevX Price: $39.99

Windows NT Security
A solid security foundation for enterprise NT 4 and Windows 2000 networks.
Paperback
By Michael McInerney
Pub: $49.99
DevX Price: $39.99

Practical Internet Groupware
Shows you how to build and deploy a variety of groupware applications.
Paperback *By Jon Udell*
Pub: $29.95
DevX Price: $23.96

MTS Programming with Visual Basic Use a single set of business objects from both a thin web client and a thin Visual Basic client.
Paperback w/CD
By Dan Mezick, Scot Hillier
Pub: $29.99
DevX Price: $23.99

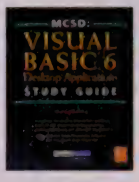
Programming Distributed Applications with COM and Microsoft Visual Basic 6.0
Execute enterprise-level development tasks formerly achievable only in the C++ environment.
Paperback w/CD
By Ted Pattison
Pub: $44.99
DevX Price: $35.99

Order Online at
http://books.devx.com

Ready-to-Run Visual Basic Code Library
An arsenal of more than 170 proven techniques and ready-to-run source code for better and more efficient programming.
Paperback w/CD
By Rod Stephens
Pub: $49.99
DevX Price: $39.99

Microsoft Mastering: Microsoft Visual Basic 6.0 Development
Work at your own pace by going step-by-step through development tasks and honing your skills through the interactive CD.
Paperback w/CD
By Teresa Canady
Pub: $49.99
DevX Price: $39.99

3-Tier Client/Server at Work
Case studies you can use—get an insider's peek and find out what went right and wrong.
Paperback *By Jeri Edwards*
Pub: $24.99
DevX Price: $19.99



MCSD: Visual Basic 6 Desktop Applications Study Guide
Provides full coverage of every exam objective. All the information you need to know!
Hardback w/CD
By Michael McKelvy
Pub: $44.99
DevX Price: $35.99

Professional Visual Basic 6 Databases
Take your programming to the enterprise level and harness the power of Visual Basic to build powerful three tier database applications.
Paperback
By Charles Williams
Pub: $49.99
DevX Price: $39.99

Inside Microsoft SQL Server 7.0
Step inside to see how system structure and behavior affect application development. **Hardback w/CD**
By Ron Soukupand, Kalen Delaney
Pub: $59.99
DevX Price: $47.99

Professional Site Server 3.0 Commerce Edition
Enter the exciting world of online business and deploy e-commerce applications on the Windows NT platform.
Paperback *By Marco Tabini*
Pub: $49.99
DevX Price: $39.99

Microsoft Agent Software Development Kit
Complete documentation and tools to deploy animated tour guides, avatars, game components--and more. usability.
Paperback w/CD By Microsoft Corporation
Pub: $49.99
DevX Price: $39.99

Visual Studio 6: The Complete Reference
Use this smart book for reference on all types of large scale projects.
Paperback w/CD
By John Paul Mueller
Pub: $49.99
DevX Price: $39.99

DevX.com carries books on these subjects:

- Access
- Active Server Pages
- ActiveX
- ADO
- C++
- Certification
- COM/DCOM
- Databases
- Dynamic HTML
- Enterprise Development
- Exchange and Outlook
- Java
- Microsoft Transaction Server
- Object Oriented Programming
- SQL Server
- Visual Basic
- Visual InterDev
- Web Development
- XML

and more!

40% OFF!
HOT SPECIALS Each Week

Order Online at
http://books.devx.com
Or call 800.848.5523 or 650.833.7100 (outside the USA)

http://books.devx.com

PRODUCTIVITY TOOLS

TOOLS FOR PROGRAMMING PROFESSIONALS

Our Best Prices On Your Favorite Apps

Developer Edition

Microsoft® Office 2000

The most efficient suite of applications for document creation, communication and business information analysis

Office 2000 extends desktop productivity to the Web, streamlining the way you work and making it easier to share, access and analyze information so you get better results. Includes Web-enabled collaboration and information sharing, rich analysis tools for better decision-making and software that is easy to use and manage. Office 2000 delivers new levels of resiliency and intelligence, enabling users and organizations to get up and running quickly, stay working and achieve great results with fewer resources.

VERSION

- Developer Edition
  Format: CD ROM
  Pub: $999
  DevX Price: $889
- Developer Edition Competitive Version Upgrade
  Format: CD ROM
  Pub: $649
  DevX Price: $589
- Developer Tools Add-On
  Format: CD ROM
  Pub: $399
  DevX Price: $369
- Bundle Standard Edition Competitive Version Canon Office Ready
  Format: CD ROM
  Pub: $388.95
  DevX Price: $299
- Professional Edition
  Format: CD ROM
  Pub: $599
  DevX Price: $549
- Professional Competitive Version
  Format: CD ROM
  Pub: $349
  DevX Price: $329
- Premium Edition
  Format: CD ROM
  Pub: $799
  DevX Price: $789
- Premium Competitive Version
  Format: CD ROM
  Pub: $449
  DevX Price: $429
- Premium w/Intellimouse Competitive Version
  Format: CD ROM
  Pub: $469
  DevX Price: $449

Microsoft® Project 98

Control your projects with the world's most popular project management software

Microsoft Project gives you the tools to coordinate responsibilities, manage resources, set up new projects and, most importantly, get your projects in on time and under budget! You'll create and edit tasks, establish and track relationships, assign resources, manage deadlines, build schedules, use charts to communicate efficiently, and share valuable information with team members. Also add audio and video to your project pages, and export project information into a database or spreadsheet. Regardless of your skill level, Microsoft Project is a must-have for all project managers.

VERSION

- Format: CD ROM
  Pub: $499
  DevX Price: $439
- Upgrade Version
  Format: CD ROM
  Pub: $199
  DevX Price: $189

SAVE $60 (full version)

SAVE $10 on your next online purchase (see page 2 for details)

Microsoft

Order Online at
http://software.devx.com

For the complete inventory of MarketPlace software, go to http://software.devx.com

Software Product Index by Publisher

Advantageware VB Advantage 3
Borland C++ Builder 5
Borland JBuilder 4, 14
Desaware Gallimaufry 3
Desaware SpyWorks 3
Desaware StorageTools 3
Desaware VersionStamper 3, 7
FarPoint Button Objx 3
FarPoint Calendar Objx 3
FarPoint Daily PlanIT 3
FarPoint Input Pro 3, 7
FarPoint List Pro 3
FarPoint Spread 3
FarPoint Tab Pro 3
IBM DB2 12
IBM VisualAge Java 14
Intara Foundation 3
KL Group JClass Chart Bytecode 3
KL Group JClass LiveTable Bytecode 3
KL Group JProbe Profiler 3, 15
Merant DataDirect Connect OLE DB Pack .. 13
Merant DataDirect ODBC 13
Microsoft FrontPage 6
Microsoft Project 98 18
Microsoft Office 2000 18
Microsoft SQL Server 13
Microsoft Visual C++ 5
Microsoft Visual InterDev 11
Microsoft Visual SourceSafe 10
Microsoft Visual Studio 10
Microsoft Visual Basic 4, 11
Microsoft Visual J++ 11
Pervasive Tango 2000 6
ProtoView ActiveX Component Suite 3, 7
ProtoView ActiveX Enterprise Suite 3, 7
ProtoView Data Explorer 3
ProtoView Data Table 3
ProtoView JFCSuite 3, 15
ProtoView JSuite 3
ProtoView ScheduleX 3
ProtoView TreeViewX 3
ProtoView WinX Component Library 3
Seagate Crystal Reports 9
Seagate Info 8
Sheridan ActiveListBar 3
Sheridan Active Suite Plus 3
Sheridan ActiveThreed Plus 3
Sheridan ActiveToolBars Plus 3
Sheridan ActiveTreeView 3
Sheridan Calendar Widgets 3
Sheridan Code Assist 3
Sheridan Components Suite 3, 8
Sheridan Data Widgets 3
Sheridan Designer Widgets 3
Sheridan VBAssist 3
Sybase Adaptive Server 12
Sybase PowerBuilder 12
Sybase PowerDesigner 12
Sybase SQL Anywhere Studio 12
Symantec VisualCafé 15
VideoSoft VS-OCX 3
VideoSoft VSFlexgrid Pro 3, 9
VideoSoft VSVIEW 3

Attention DevX.com Premier Club Members

Don't forget your member discount!

10% OFF DOWNLOAD SOFTWARE EVERY DAY

Members get extra savings over and above our everyday great prices on downloadable software.

Not a member?
Join today at www.devx.com

Printed in the U.S.A. The Development Exchange, DevX, and DevX.com are trademarks of Fawcette Technical Publications. All products listed are registered marks or trademarks of their respective owners. DevX.com MarketPlace online and catalog prices are subject to change without notice. © 1999 Fawcette Technical Publications. All rights reserved.

Order Online at

http://software.devx.com

Or call 800.848.5523 or 650.833.7100 (outside the USA)

Customer Information

Ordering and Shipping Information

- Order at **http://marketplace.devx.com**, or call **1-800-848-5523 (outside the US call 650-833-7100)**
- To access the book and software stores directly go to **http://software.devx.com** or **http://books.devx.com**
- For additional questions regarding shipping policies and fees, please go to our online Customer Service area, or call our Customer Service Department at **1-800-848-5523 or 1-650-833-7100**

DevX.com MarketPlace Exclusives!

VBCD

Your complete guide to *Visual Basic Programmer's Journal* conveniently located on one CD. This monthly CD features archives of *VBPJ* (1996-current), including code, and links to online VB resources. An invaluable aid for anyone who needs a comprehensive library of programming information, the VBCD is your portable resource to take wherever you need it.

Bonus Offer: Your subscription also includes a separate one-year subscription to the Premier Club of DevX.com, a $44.95 value.

DevX Price: $99.95 for 12 issues

(Canada/Mexico add $8 for shipping, International add $15)

Web Builder CD

The best tools and technologies to help build better and faster Web sites

Web Builder CD-ROM has over 300 pages of how-to articles, must-have tools and utilities, expert solutions, plus sample code to get you up and running.

It's everything you need to get your Web site started quickly and easily, including HTML editors, GIF animators, site statistic tracking software, Web servers and more. You'll be up and running with expert guidance from the leading publishers on Windows and Internet programming.

Newsstand Price/4 issues: $51.80
DevX Price: $29.95, 4 quarterly issues

(Canada/Mexico add $8 for shipping, International add $15)

http://software.devx.com • http://books.devx.com

DEVX
The Development Exchange

The Development Exchange
209 Hamilton Ave.
Palo Alto, CA 94301

Order Online and Save!

YOUR CEO WANTS RELIABILITY, AVAILABILITY AND SECURITY.
YOUR CIO WANTS HIGH PERFORMANCE.
YOUR COO WANTS BROAD FLEXIBILITY.
YOUR CUSTOMERS WANT IT YESTERDAY.

HERE'S WHAT YOU WANT.

Our customers wanted all of the above, too. And they got it with Sybase Enterprise Application Server.™ Check out the success of Blue Cross/Blue Shield, First American, and others at www.sybase.com/easerversuccess. Or call Sybase at 1-800-8-SYBASE. And make everybody happy. Especially yourself.

SYBASE®
INFORMATION ANYWHERE.™

©1999 Sybase, Inc. All rights reserved. All trademarks are the property of their respective owners.

way to implement this message structure. XML was originally designed to be the data format for exchanging structured data over the Web. For that reason, we chose to implement our messages in an XML format. A sample message from the JProxyApplet to the ASP page and a message from the ASP page to the JProxyApplet would look like the samples in Listing 1.

The next step was to define the lowest point of communication on both the client and the server. Specifically, on the client we built the JProxyApplet, and on the server we developed the ASP page. To start with the server piece, the ASP page residing on the IIS server, we chose to have only one input: the message itself. Therefore, incoming requests are made to the ASP page with the following format:

```
http://mycompany.com/services/
  components.asp?message=<methodcall>
```

The parameter "message" contains the entire method call request in the formatted XML that we'd already defined. From within the ASP page, we then call the business object that will parse the XML and perform the method calls.

On the client, we designed the JProxyApplet (see Listing 2), a small, invisible applet that listens for messages from client applications and then forwards those messages on to the ASP page residing on an IIS server. First, the JProxyApplet asks the virtual machine for both connect and listen privileges by calling the Java Security API. The security calls will be slightly different depending under which browser the applet runs. The applet can detect which browser is hosting the applet and then set the necessary privileges.

Next, the applet creates a ServerSocket object, which will listen on a predefined port on the local machine. After defining the ServerSocket, a socket can be initialized to accept all messages on that port. Using the Socket object, along with Java's built-in streaming capabilities, we can send and receive messages. Once we have set up the socket, we wait indefinitely for data from the client application.

Whenever a message arrives on the socket, within the JProxyApplet, we create a new URL object and POST the message to the ASP page. The results of the POST are then sent over the socket's output stream and back to the caller—in this case, the Java client. Since Java has strong networking capabilities built into the VM, all the communication can be handled in fewer than 100 lines of Java code.

The final piece of the architecture is the ActivatorObject, the business object that activates COM+ components (see Listing 3). The ActivatorObject resides on the same server as the ASP page. The ASP page calls the ActivatorObject, passing in the XML message received in the query string parameter. From there, the ActivatorObject parses the XML. We used the Microsoft XML parser, since we were creating the ActivatorObject in Visual Basic, and Microsoft's XML DLL made it easy to manage and manipulate our data. Once the XML message has been parsed, we use the CallByName() method in VB to activate the COM+ object and pass in the necessary parameters. Any return values from the COM+ method calls are returned as XML strings. The ASP page then takes the XML return string and passes it back to the JProxyApplet by issuing a Response.Write(). It should be noted that the only text going back across the wire to the JProxyApplet is the XML message itself. In other words, there are no HTTP headers or tags returned in the contents of the response.

Since the XML messages are defined rather generically, we can pass a wide variety of data types to and from our COM+ components, including strings, numerics, booleans, and even ADO Recordsets. In the special case of ADO Recordsets, we persisted the Recordset to XML using the ADO 2.1 Save() method (with the next version of ADO, Recordsets can be converted directly from an object to XML, which would eliminate this step). In both the JProxyApplet

Get FREE news, tips, and article updates from DevX.com

DevX has expanded its e-mail newsletters to give you a choice of information targeted to your specific programming needs. Choose specific topics or receive information on all areas.

Newsletters focus on nine special interest areas, including:

- Weekly DevX HotLinks News (HTML format)
- XML Update
- VB Update
- Java Update
- Enterprise Update
- SQL Update
- Web Development
- C++ Update
- MarketPlace Specials
- ASP Update

DEV·X The Development Exchange

It's Free. Sign up today at www.devx.com

Fawcette Technical Publications • 209 Hamilton Avenue • Palo Alto, CA 94301 • 650-833-7100 • www.fawcette.com

DevX is a trademark of Fawcette Technical Publications. Java is a registered trademark of Sun Microsystems. All other product names are the properties of their respective owners.

Listing 2 Walking through the Code The JProxyApplet is a small, invisible applet that listens for messages from client applications and then forwards those messages on to the ASP page residing on an IIS server.

```
Step 1: Enable connect and listen privileges (Netscape only)
PrivilegeManager PM =
  PrivilegeManager.getPrivilegeManager();
PM.enablePrivilege("UniversalConnect");
PM.enablePrivilege("UniversalListen");

Step 2: Create and open a socket
server = new ServerSocket(9801);
socket = server.accept();

Step 3: Setup the input and output streams
is = socket.getInputStream();
os = socket.getOutputStream();

Step 4: Wait for incoming data
while (is.available() == 0)
{ /* wait */ }


Step 5: POST the data to the ASP page
String strURL = new String(
  "http/mycompany.com/services/components.asp?message=" +
  strMessage);
URL url = new URL(strURL);

Step 6: Wait for return data
BufferedReader in = new BufferedReader(
  new InputStreamReader(url.openStream()));
while ((current_line = in.readLine()) != null)
{
  output = output + current_line;
}
strResponse = output.toString();


Step 7: Send the data over the socket and back to the client
BufferedWriter outputBuffer = new BufferedWriter(
  new OutputStreamWriter(os));
outputBuffer.write(strResponse);
outputBuffer.flush();
```

Listing 3 ActivatorObject The ActivatorObject is the business object that activates COM+ components.

```
Step 1: Load the XML string into the Microsoft XML parser
Dim objXMLDoc As New MSXML.DOMDocument
Call objXMLDoc.loadXML(XMLString)

Step 2: Parse the contents of the message
Set objNode = objXMLDoc.getElementsByTagName("progid")

Step 3: Call the component's method
Set vntOut = CallByName(objGeneric, strMethod, VbMethod,
  vntParameters)

Step 4: Return the results
PerformMethod = "<returnvalue>" & vntOut & "</returnvalue>"
```

and the ActivatorObject, logging and diagnostic code was added to allow for more complete tracking of messages within the system.

Learning from the Architecture

This architecture presents you with several benefits and some drawbacks. One big benefit: Developers have full control of the underlying workings of the system. For example, changing the message layout to support additional functionality is simple. On a related note, changing the message structure of the communication mechanism will not inherently break either the client or server code. Since a "proxy" piece is used on both the client and server side of the communication, both the client and the server can be shielded from the interface (message structure) change by handling the additional structure elements within the JProxyApplet and ActivatorObject pieces.

This architecture also gives you the benefit of scalability. Since the server is a Web server, implementing a Web farm to increase the number of users the application can support is easy. And if developers build stateless components using COM+ component services, the architecture will continue to support an increasing number of users over time. In addition, COM+ adds load-balancing capabilities, which will lower the amount of redundancy and fail over code that needs to be written for the ActivatorObject, and should ease administration of multiple business components across multiple servers.

An interesting twist to this architecture is security. While integrated NT security will no longer be available for the COM+ components running under this architecture (they will now be activated from an ASP page under the IUSR user account), the architecture will allow for client authentication using client certificates and SSL. In other words, there is a shift in security models from an integrated NT security model to an Internet certificate/SSL model. Similarly, encryption is another benefit of shifting to the certificate/SSL model, since you get encryption "for free" with SSL.

Finally, because the architecture relies heavily on industry-proven protocols such as HTTP, HTTPS, sockets, and SLL, it is likely to be supported over the next decade. While these standards may be modified, they'll still be a part of the distributed application landscape. HTTP and XML are both well-supported, which makes it easy to develop solutions that implement both protocols.

Obviously, the environment will determine what approach you should take for developing cross-platform or cross-development tool applications. The level of interoperability you need and the level of complexity you want to manage will determine which approach works in your environment. If you already have Web servers in place and want to leverage your existing COM+ components from different systems, the approach may work for you. If you want to take advantage of the new features of COM+ without dropping your existing non-Microsoft applications, the architecture offers one way of achieving a high level of cohesion.

This technique is just one way of sharing data between applications. Examine other approaches in light of your needs before making an infrastructure change. Depending on the strictness of your organization's standards, your budget, and third-party software availability, a solution can be implemented. The key is developing or deploying the right tools—tools that do not exceed the budget or sacrifice functionality. And you must try to anticipate the need for interoperability. If you do enough planning, you'll be better prepared to implement a solution that is cost-effective and appropriate for your environment. ■

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0002 Download all of the code for this issue of *Java Pro*. Free online registration is required.

JP0002JB DevX Premier Club members download the code for this article.

JP0002JB_T DevX Premier Club members can read this feature online.

Not a member yet? Go to www.DevX.com to sign up for the free Registered Level or to subscribe to the Premier Club.

Javatecture

How I Started Using JavaServer Pages

I've spent some time recently trying to understand JavaServer Pages (JSP) and why and where you use them. They're a clever technology, although they do result in messy pages of code.

You'll remember that we've discussed Java servlets—a Java technology to replace cgi-bin client-server programming with a more efficient model. Ordinary cgi-bin programs are launched by the Web server when HTML form pages are sent to a particular directory. These cgi-bin programs parse the form information and create a new dynamic HTML response page, which they return to the client. Java servlets improve on this methodology by keeping a servlet engine running, which catches cgi-bin programs and parses them in Java. The servlet then computes the desired response page and returns it to the client in much the same way.

The problem with both of these approaches is that there is no good way to construct complex output HTML pages. Usually, programmers rely on some kind of template pages, which they copy and fill in with data from a particular cgi-bin query. This is one reason why most response pages look simple, and many are downright ugly.

JavaServer Pages are one solution to this problem. A JSP page contains both HTML to create the final output page and Java to fill in the parts that must be computed. It also does the form parsing directly and almost invisibly for you. JavaServer Pages are supported by a class of programs called application servers. Some of these packages add on to existing Web servers; others perform both the Web server and the JSP functions directly. You will find a complete list of products at java.sun.com/products/jsp. You should be careful when ordering these products, because some of them support only back-level versions of JSP (0.91 was popular, but quite different) and the syntax of JSP pages has changed substantially in the final 1.0 release.

JavaServer pages let you interweave Java and HTML—this is both a strength and a weakness

BY JAMES W. COOPER

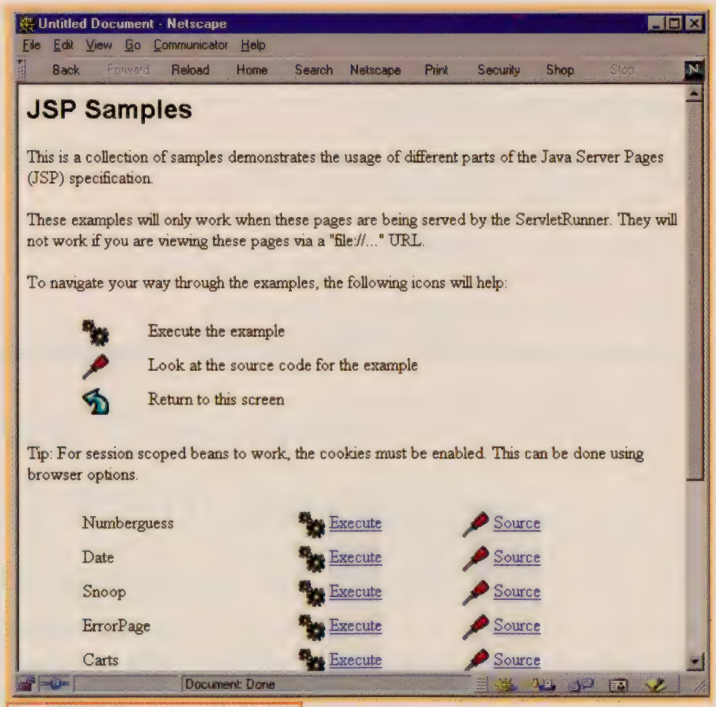

Untitled Document - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
JSP Samples
This is a collection of samples demonstrates the usage of different parts of the Java Server Pages (JSP) specification.
These examples will only work when these pages are being served by the ServletRunner. They will not work if you are viewing these pages via a "file://..." URL.
To navigate your way through the examples, the following icons will help:
Execute the example
Look at the source code for the example
Return to this screen
Tip: For session scoped beans to work, the cookies must be enabled. This can be done using browser options.
Numberguess Execute Source
Date Execute Source
Snoop Execute Source
ErrorPage Execute Source
Carts Execute Source
Document: Done

Figure 1 JSP Samples Page You can learn a lot from these examples by watching them run and by examining the code.

Installing the JSP Development Kit

You will also find the complete JSP development kit at the Web site under the name JSWDK-1.0. If you download this package and install it, you will have a complete working JSP system where you can develop and test JSP pages as well as servlets and then deploy them to your site. The JSP engine provided in this kit is adequate for small sites, but for larger sites, you need one of the application servers. An application server is a superset of the servlet engine and supports both JSP pages and ordinary servlets.

To try out the servlet engine on Windows NT, go to the JSWDK-1.0 directory and run startserver.bat. On Unix machines, run the startserver shell script that also is provided. I did all my testing on Windows NT 4.0 with Service Pack 5 installed. I wrote my test programs on a laptop that was not connected to any network. I was not able to get the examples to run on the same laptop running Windows-95b (OSR2).

Listing 1 Connecting with Beans Most JavaServer Pages connect with simple JavaBeans that contain and return the contents of the fields on the page. The beans that JSP pages use are invisible server-side beans that have just get() and set() methods. They have no visual properties.

```
package mine;
//A simple Java bean for the Hi demonstration programs
public class JTest {
  private String text; //text stored here

  public JTest() {
    text = null;
  }
  //set text here
  public void setEntryText(String t) {
    text = t;
  }
  //get it back here
  public String getEntryText() {
    return text;
  }
}
```

To see the test programs that come with the JSWDK, start your Web browser and enter the URL http://localhost:8080. This brings up a page describing the available examples and the servlet API. If you click on the JSP Samples link, you will get the page shown in Figure 1. You can learn a lot from these examples, by watching them run and examining the code. However, since JSP pages are still poorly documented, we'll start at the beginning with a simple "Hi" program and work up to some of the concepts they illustrate.

A Hello World JSP Program

A JavaServer Page consists of HTML, JSP directives, and Java code. The JSP engine compiles this code and caches it when you first access the page. This makes further accesses to the page much faster. The JSP page is a server-side concept: the code that is sent back to the client is pure HTML generated by the code on that page. This makes the client requirements much less demanding than those that require particular levels of Java support.

Our first simple program, hi.jsp, consists of a simple form, some output, and communication with a back-end JavaBean. All JavaServer Pages must start even before the <html> tag with declarations defining that it is a JSP and the JavaBean it will communicate with:

```
<%@ page language="java" import="mine.JTest" %>
```

or

```
<jsp:useBean id="mine" scope =
  "page" class="mine.JTest" />
```

The first line is a jsp directive, describing the language (only Java is supported) and the bean the page will communicate with. The second line is the same jsp directive in an alternate format. You need only one or the other, although both do no harm. Note that the longer form follows the XML custom, terminating the tag with a /> to indicate that nothing more follows that relates to that tag.

JavaServer Pages are a clever technology, although they do result in messy pages of code

JavaBeans and the JSP

Most JavaServer Pages connect with simple JavaBeans that contain and return the contents of the fields on the page. The beans

TechTIP

Center Your Frames and Dialogs

You can find many tips on how to center a window within the screen, but how about centering a window in its parent container? The following code shows you how to do that. Say you have a JFrame (or Frame), and you launch a JDialog (or Dialog) like this:

```
JDialog jd = new JDialog(this);
//you have to pass the this pointer of Frame to establish
//parent-child (or owner-owned) relationship
jd.setVisible(true);
```

In the code for JDialog, you can use the following to center the dialog with respect to its owner frame:

```
public void centerInParent ()
{
  int x;
  int y;
  Container parent = this.getParent();
  Point topLeft = parent.getLocationOnScreen();
  Dimension parentSize = parent.getSize();
  Dimension ownSize = this.getSize();
  if (parentSize.width &lt; ownSize.width)
    {x = ((parentSize.width - ownSize.width)/2) +
      topLeft.x;}
  else
    {x = topLeft.x;}
  if (parentSize.height &lt; ownSize.height)
    {y = ((parentSize.height - ownSize.height)/2) +
      topLeft.y;}
  else
    {y = topLeft.y;}
  this.setLocation (x, y);
  this.requestFocus();
}
```

For the sake of completeness, the following code centers a Dialog or Frame within the screen:

```
public void centerInScreen()
{
  Dimension dim = this.getToolkit().getScreenSize();
  Rectangle bounds = this.getBounds();
  this.setLocation((dim.width - bounds.width) / 2,
    (dim.height - bounds.height) / 2);
  this.requestFocus();
}
```

Behrouz Fallah

Figure 2 Unsatisfactory Results The simple hi1.jsp program (a) before and (b) after entering a name and pressing submit. This is not satisfactory, because we display "Hello there: null" before any name has been submitted.

used by JSP are invisible server-side beans that have just get() and set() methods. They have no visual properties. For example, you might have a <form> section on your page and inside it a text entry field.

```
<input type="text" name="entryText" size="25">
```

The associated JavaBean then has get() and set() methods for this field (see Listing 1). Note that the case is odd but required. The name of the field must start in lower case, but the get() and set() methods convert it to upper case. When you begin using this bean on a JSP, you must initialize all of its properties to access them.

```
<jsp:setProperty name="mine" property="*"/>
```

This line initializes all the bean's properties at once. This statement is required, or your bean properties won't work.

Communicating with the JavaBean

The complete form section of our JavaServer Page looks like this:

```
<form method="post">
<input type="text" name=
 "entryText" size="25">
<br>
<input type="submit" value="submit">
</form>
```

Note that although a post() method is specified, no *action* is specified that tells the page where to send the form. Instead, submit just sends all the form data to the JavaBean, executing the set() method for each form field. In this case, the JSP calls the bean's setEntryText() method with the string from the entry field as its argument. Thus, following the submit, the bean contains all the data in the form's entry fields.

You can access that data using the get() methods and can print it out on the Web page using JSP statements:

```
<%
out.println(mine.getEntryText());
%>
```

Note that you can embed real Java statements on a JavaServer Page as long as they are enclosed in the <% and %>. You can also use a shorter form, where the equals sign replaces the "out.println" method call:

```
<%= mine.getEntryText() %>
```

This form is not a complete Java statement and does not need a terminating semicolon. Since it is so compact, you see it frequently.

Listing 2 A Complete Page This program initializes the *mine* bean, uses the submit button to load it, and prints out the result.

```
<jsp:useBean id="mine" scope =
 "page" class="mine.JTest" />
<jsp:setProperty name=
 "mine" property="*"/>

<html>
<head><title>Hi</title></head>
<body>

<form method="post">
<input type="text" name=
 "entryText" size="25">
<br>
<input type="submit" value="submit">
</form>

Hello there: <%= mine.getEntryText() %>

</body>
</html>
```

Listing 3 Our Hello World Program This example tests for whether the name is non-null before displaying it.

```
<jsp:useBean id="mine" scope =
 "page" class="mine.JTest" />
<jsp:setProperty name=
 "mine" property="*"/>

<html>
<head><title>Hi</title></head>
<body>

<form method="post">
<input type="text" name=
 "entryText" size="25">
<br>
<input type="submit" value="submit">
</form>

<% if (mine.getEntryText() != null) { %>
 Hello there:
 <%= mine.getEntryText() %>
<% } %>

</body>
</html>
```

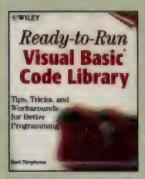
Avoid obsolescence. Visit ComputerJobs.com today and extend the life of your career. | **computerjobs.com**
Upgrade your future.

DEV·X
The Development Exchange

MarketPlace
BOOKS

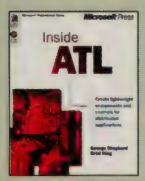
Here's just a glimpse of the books in the DevX bookstore...

Sams Teach Yourself Java 2 Online in Web Time
Paperback w/CD
By Stephen Gilbert and Bill McCarty
Pub: $49.99
DevX Price: $39.99

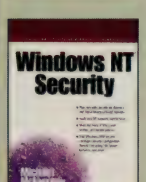

Ready-to-Run Visual Basic Code Library
Paperback w/CD
By Rod Stephens
Pub: $49.99
DevX Price: $39.99

Hitchhiker's Guide to VB & SQL ServerSixth Edition
Paperback w/CD
By William R. Vaughn
Pub: $49.99
DevX Price: $39.99

Dynamic HTML in Action
Paperback w/CD
By Eric M. Schurman and William J. Pardi
Pub: $39.99
DevX Price: $31.99


Inside ATL
Paperback w/CD
By George Shepherd and Brad King
Pub: $49.99
DevX Price: $39.99

Building Corporate Portals with XML
Paperback
By Clive Finkelstein and Peter Aiken
Pub: $49.99
DevX Price: $39.99


Windows NT Security
Paperback
By Michael McInerney
Pub: $49.99
DevX Price: $39.99

40% OFF!
HOT SPECIALS Each Week

Order Online at
http://books.devx.com
Or call 800.848.5523
or 650.833.7100 (outside the USA)

DevX.com MarketPlace prices are subject to change without notice. The Development Exchange, DevX, and DevX.com are trademarks of Fawcette Technical Publications. All products listed are registered marks or trademarks of their respective owners.

http://books.devx.com

Listing 4 Diminishing Clutter In this example, the test for printing Hello there and a name are moved to another file, which keeps the HTML and server-side logic together, but logically separated. This approach gives the Web designer control over the base jsp file and the server programmer control over the output in the myResponse.jsp.

```
<jsp:setProperty name="mine" property="*"/>

<html>
<head><title>Hi</title></head>
<body>

<form method="post">
<input type="text" name="entryText" size="25">
<br>
<input type="submit" value="submit">
</form>

<%@ include file="myResponse.jsp" %>

</body>
</html>
```

Our First Complete JavaServer Page

Combining these fragments, we have a complete page (see Listing 2). This program initializes the *mine* bean and uses the submit button to load it. Then it prints out the result.

The displays in Figure 2 are not satisfactory, because we display "Hello there: null" before any name has been submitted. It would be better to display this only when there is real data. The second JSP example accomplishes this by testing for whether the name is non-null before displaying it (see Listing 3).

In this example, we do not see any "Hello there" text or null name displayed. Note how we can interweave Java and HTML in the same page to make decisions about what to display. This feature is both the greatest strength and weakness of JSP pages. You can write Java code to display data, but interweaving it makes the page look unstructured and confusing.

Now the purpose of JSP is to keep HTML and server-side logic together, but logically separated to diminish clutter. The final example, hi3.jsp (see Listing 4), just moves the test for printing Hello there and a name to another file, which is referred to using an include statement.

This gives the Web designer control over the base jsp file and the server programmer control over the output in the myResponse.jsp, which is much the same code as in hi2.jsp:

```
<% if (mine.getEntryText() != null)
{ %>
 Hello there:
 <%= mine.getEntryText() %>
<% } %>
```

Where to Put All These Classes?

If you install the JavaServer Page JDK as we described, you will have a tree that contains

```
Jswdk-1.0\examples\jsp
```

along with many little directories underneath for each of the provided examples. Unzip the jsp code into a new "hi" folder at this level.

The JavaBean goes into the hierarchy:

```
Jswdk-1.0\examples\Web-inf\jsp\beans
```

Create a folder called "mine" and unzip the bean into it.

Automated Tools

You can see the power of JSP compared to other approaches. However, you can also see the clutter and confusion they can generate. The newest versions of several of the Java IDEs, including Borland's JBuilder and IBM's Visual Age for Java, contain methods for building JSP a little more automatically. These newer generations of IDEs should help developers reduce the clutter and unleash the powerful potential of JavaServer Pages.

James W. Cooper is a computer science researcher and the author of 13 books. His latest, on Java design patterns for Addison Wesley Longman, is due out this month.

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0002 Download all of the code for this issue of *Java Pro*. Free online registration is required.
JP0002JT DevX Premier Club members download the code for this article.
JP0002JT_T DevX Premier Club members can read this feature online.

Not a member yet? Go to www.DevX.com to sign up for the free Registered Level or to subscribe to the Premier Club.

Unit Testing Prevents Errors

by Adam Kolawa

In previous columns, we have focused on coding standards. This month, we will introduce unit testing as a method of error prevention. To understand the importance of unit testing, we must first take a look at the state of testing in software development today.

Developers today face increasing deadline pressure as they write code in a competitive software market. They often write code quickly without regard for quality, and then send it off to another department to have it tested. However, the best person to test a segment of code is the developer who wrote it. He or she understands the code better than anyone else and is most capable of finding the weaknesses in the code.

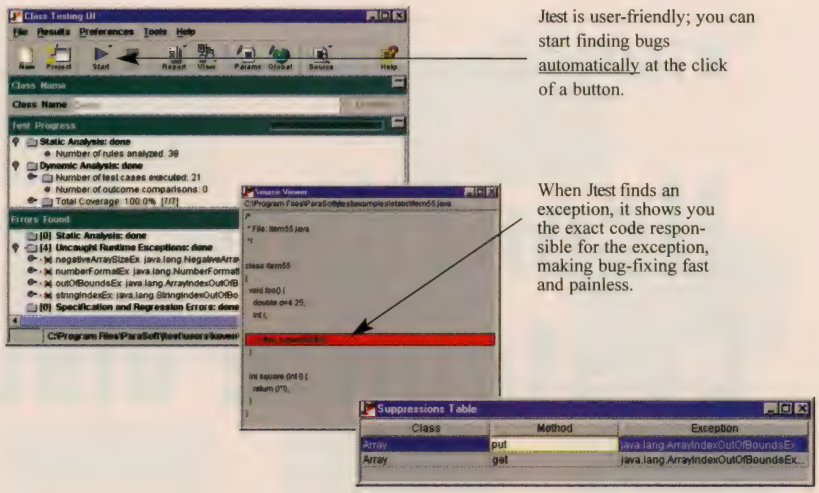
As developers test code, they should not just be trying to finish quickly and go home early. Rather, they should be trying to break the code in any way possible. Successful testing is not merely any testing that finds no errors; it is testing that flushes out errors. Instead of mentally rewarding yourself for code that appears clean, you should reward yourself for finding bugs. Just because code compiles cleanly does not mean that the code is ready to ship to customers. Adopt a new, more critical outlook on your code, and you will see dividends in the cleanliness of that code.

Once you have adopted this new outlook towards testing, you need to find an effective method of testing. One of the most effective methods is unit testing.

Often, developers hear about unit testing and think the term refers to module testing. But the term "unit testing" refers to an even lower level of testing--a testing that occurs at the class or object level. Class-level testing is extremely important to the well-being of any application, but few developers perform class testing. Most developers think of class testing as a sort of "golden rule"--a form of testing that they would perform in an ideal world if they had time. However, because of the labor and mechanics involved, they avoid class testing and believe that it is not possible to perform on large applications.

In upcoming columns, we hope to illustrate that although unit testing at the class level involves some extra work when it is done by hand, the technique is certainly feasible and worthwhile. Furthermore, technology has evolved to the point that developers can now choose from a number of powerful tools that make unit testing as easy as breathing.

Dr. Adam Kolawa is President and CEO of ParaSoft.
He can be reached at ukola@parasoft.com



Jtest is user-friendly; you can start finding bugs automatically at the click of a button.

When Jtest finds an exception, it shows you the exact code responsible for the exception, making bug-fixing fast and painless.

With Jtest, it's easy to focus on the exceptions that are most important to you. You can customize reports to suppress exceptions by type, name, and enclosing method.

Would you believe that Jtest® can take your brand-new class and test it on its own, automatically?

*"Jtest® represents a dramatic step forward in Java™ testing. As developers move toward serious application development, it is absolutely critical for them to incorporate tools such as Jtest in their development cycle." -- **Will Iverson**, Symantec*

Jtest® is the world's first automatic class testing tool for Java™. Testing on its own with one click from the developer, Jtest automatically passes unexpected inputs to each class and validates that the inputs will not cause the program to crash. Using patented Test Generation System Technology (#5,784,553 & 5,761,408), Jtest tests all methods of a class. When Jtest finds an input that generates an uncaught runtime exception, it reports to the developer the arguments that cause the problem, the stack trace for the exception found, and the calling sequence that shows the exception happening. The testing process with Jtest prevents errors that could cause problems when classes are integrated into modules.

For version 3.0, ParaSoft has added three impressive new testing capabilities to Jtest: black-box testing, regression testing, and static analysis.

Black-box testing, which ensures that each class behaves according to specification, is now a quick and easy process with Jtest. During black-box testing, Jtest automatically provides a set of inputs based on sophisticated analysis and executes the class with the inputs. The user can also provide a set of inputs that will be tested along with the automatically generated inputs. Jtest will then report the outcomes, which the user can verify with one click of the mouse.

Jtest automates regression testing, a process similar to black-box testing. During regression testing, Jtest helps make sure that developers do not re-introduce errors to their code. Jtest remembers outcomes from previous runs and compares outcomes every time the class is tested, reporting an error if an outcome changes. Developers can use Jtest to make sure they are not letting their code regress into previous errors when they make changes.

In addition, you can now enforce Java coding standards automatically with the new static analysis feature of Jtest. Jtest parses the Java source of a class and applies to it a set of rules. Any violations to these rules are reported as errors. Enforcing coding standards at every stage of the development process will lessen the amount of debugging developers must perform later in the development cycle.

If you would like the satisfaction of using Jtest, visit www.parasoft.com/superior.htm or call ParaSoft toll-free at (888)305-0041 for more information.

ParaSoft®

www.parasoft.com/superior.htm

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. ParaSoft is a registered trademark and Jtest is a registered trademark of ParaSoft Corporation.

Visual Components

JIconEditor Lets Users Customize Visual Elements

Supporting customizability adds value to graphics applications and IDEs

BY CLAUDE DUGUAY

One of the best ways you can add value to your application is by supporting a high degree of customizability. Swing gives you an immediate edge, with a pluggable look-and-feel model that lets you offer flexibility in only a few lines of code. An interesting enhancement is to allow users to customize visual elements such as icons. Depending on your customers, this may be more than you want to offer, but in some applications, such as integrated development environments or graphic applications, this is a nice feature. This month's visual component is a full-function icon editor.

Figure 1 shows JIconEditor in action. As you can see, we support common tools such as a pencil for single pixel editing; an eye dropper for sampling colors; paint fill, erase, move, and shape drawing tools; and vertical and horizontal flipping and rotation.

We'll use several classes in this component; you can see how they relate to each other in Figure 2. JIconEditor uses three toolbars derived from the ToolPanel class, each of which uses a specific type of button. The PixelEditor is an extension of a PixelView and uses an ImageOps class to delegate image operations. The Action and Draw buttons and toolbars, along with the PixelEditor, make use of some common constants declared in ActionConstants and DrawConstants interfaces for convenience. Not all of these classes can be listed in print, so be sure to check www.java-pro.com to get the full listings.

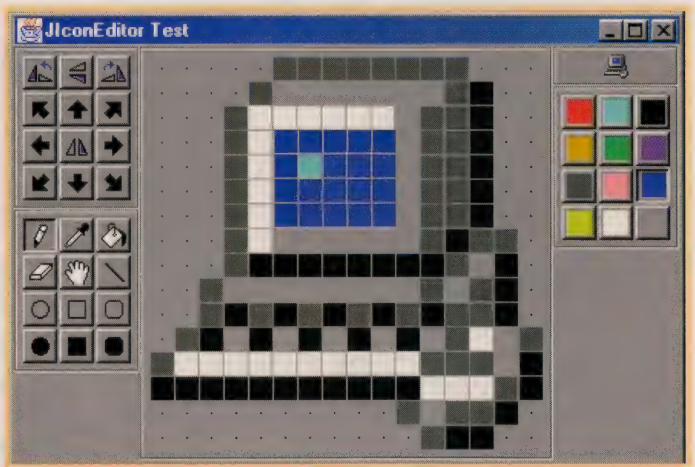


Figure 1 JIconEditor in Action JIconEditor supports familiar tools.

Toolbars

Rather than use dockable JToolbar implementations, I chose to use a base class called ToolPanel, which is primarily a JPanel intended to support action event dispatching. ToolPanel implements addActionListener(), removeActionListener(), and fireActionEvent() methods, which allow us to send notification to listeners whenever something happens. If you want to make these dockable, you can always drop them into a JToolBar without making any sacrifices.

The three toolbars in our implementation are shown in Figure 3: the DrawToolbar is responsible for selecting drawing tools; the ActionToolbar triggers actions like flip or move; and the PaletteToolbar lets you select the active color. (If you read last month's column, you know I was distracted into developing a JPalette component for use in this context. Here I'm intentionally implementing a simplified palette toolbar so that there are no code dependencies between columns. If you download the code from both columns, you can substitute the PaletteToolbar with the JPalette to produce an even more sophisticated icon editor.)

The PaletteToolbar uses a PaletteButton (see Listing 1), which extends JToggleButton so that we can make these members of a ButtonGroup. This allows us to create a mutually exclusive selection group without much effort. The PaletteButton class stores a color, which we make accessible from a getColor() method. We also set up a few parameters like the preferred and maximum size, image positioning, and focus handling (which we effectively disable). The icon is built in the constructor with a BufferedImage, filled with the selected color.

Since we want each of the buttons in the toolbar to be added to the panel, registered with an action listener, and be a member of a ButtonGroup, we'll construct them in advance in an array and iterate through the array, operating on each in the same way. The only other thing our constructor does is set up a GridLayout and a suitable Border.

The PaletteToolbar class (see Listing 2) implements the ActionListener interface and receives notification when a button is pressed. From that, we store the active color and fire an action event of our own to any ActionListener registered with the PaletteToolbar. This is easy, since we're inheriting from the ToolPanel and need only call the fireActionEvent() method. You'll see this pattern repeated in the rest of our toolbars.

The DrawToolbar class implements another set of mutually exclusive buttons. These DrawButton instances determine the drawing context we are in, such as erasing or drawing pixels or shapes. To make it easier to reference resources, we use a DrawConstants interface that lets us include the same constants in other classes. If you've never seen this technique before, it's convenient primarily because you don't have to keep qualifying each constant with the class in which it was declared, since the interface can be included anywhere.

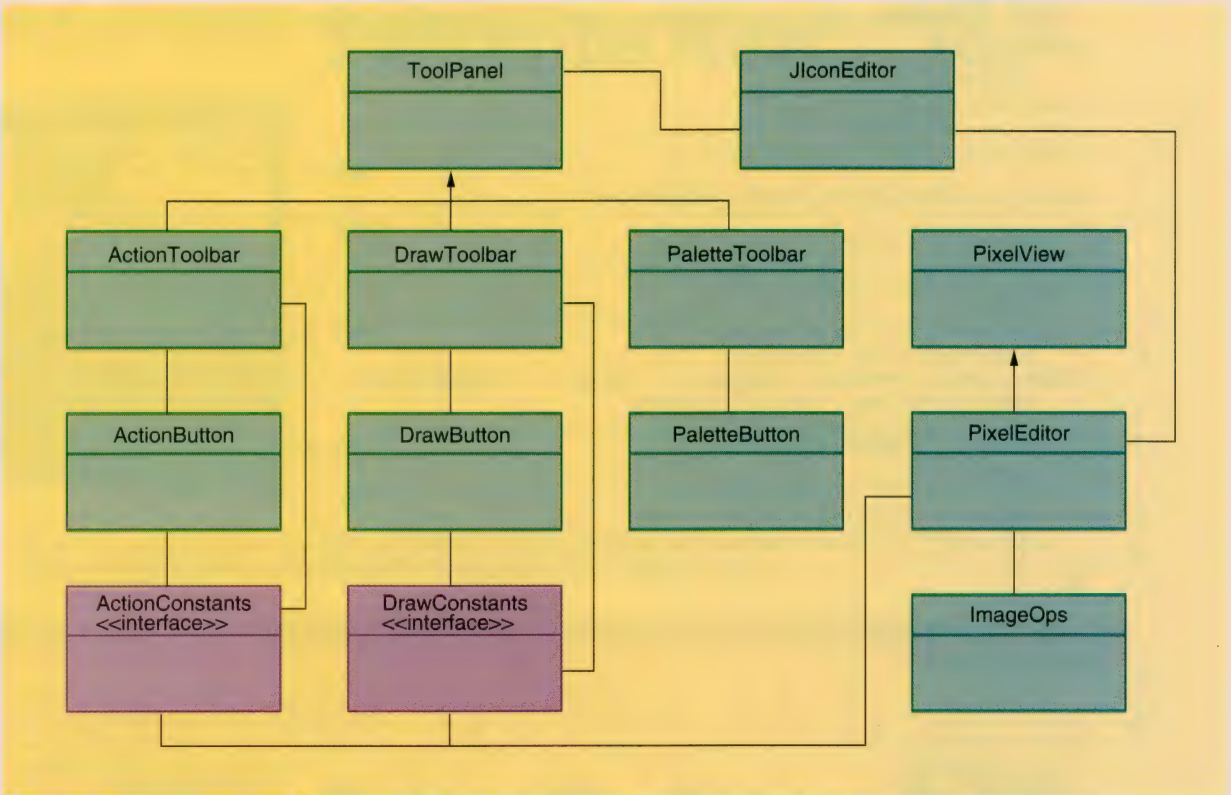


Figure 2 Class Relationships JIconEditor includes the following classes. It uses three toolbars derived from the ToolPanel class, each of which uses a specific type of button.

The DrawConstants serve a dual purpose. The constants remove any tight coupling to specific values and let us change the values without affecting behavior. They also store the prefix names for the icon files we intend to use. These are loaded by the DrawButton class, along with a suitable cursor image.

If you've never defined custom cursors, it's easy. The trick under Windows is to use the upper left 16×16 pixels in a 32×32 pixel image. Just use a GIF file and make the background transparent. This may work differently on other platforms according to the API documentation, but this technique was not obvious and required some experimentation. You can create custom cursors with the createCustomCursor() method in the AWT Toolbox class.

We use an ImageIcon to store both the Icon and Cursor images in DrawButton (see Listing 3). The preferred size and icons are set up in the constructor, along with a command name and a reference to the cursor for later use. We can get the cursor with a call to getToolCursor(). Be careful not to rename this to getCursor(), or you'll override the component-assigned cursor accessor and end up with confusing results. The only other thing to keep in mind with cursors is the hot spot, which is defined as specific pixel position for the pointer, typically the tip of an arrow or the bottom of an eye dropper, for example.

We define the hot spot as a point argument in DrawButton and set things up in the DrawToolbar class. Like PaletteToolbar, we use an array of constructed DrawButton elements and iterate through them to add, group, and register each button. DrawToolbar does some minor setup work, like setting the border and layout manager, and watches for button action events. When a button is pressed, we fire off an action event through the fireActionEvent() method inherited from ToolPalette.

The final toolbar is the ActionToolbar class, which uses the ActionConstants interface and ActionButton elements. These buttons take immediate action and are not mutually exclusive, so they

Figure 3 The Three Toolbars These are the three toolbars used in our implementation. The (a) DrawToolbar is responsible for selecting drawing tools; the (b) ActionToolbar triggers actions like flip or move; and the (c) PaletteToolbar lets you select the active color.

Listing 1 PaletteButton Class The PaletteToolbar uses a PaletteButton that extends JToggleButton so that we can make these members of a ButtonGroup.

```
import java.awt.*;
import java.awt.image.*;
import javax.swing.*;

public class PaletteButton extends JToggleButton
{
  protected Color color;

  public PaletteButton(Color color)
  {
    setPreferredSize(new Dimension(22, 22));
    setMaximumSize(new Dimension(22, 22));
    this.color = color;
    BufferedImage image = new BufferedImage(
      16, 16, BufferedImage.TYPE_INT_RGB);
    Graphics g = image.getGraphics();
    g.setColor(color);
    g.fillRect(0, 0, 16, 16);
    setIcon(new ImageIcon(image));
    setFocusPainted(false);
    setHorizontalAlignment(CENTER);
    setVerticalAlignment(CENTER);
  }

  public boolean isFocusTraversable()
  {
    return false;
  }

  public boolean isDefaultButton()
  {
    return false;
  }

  public Color getColor()
  {
    return color;
  }
}
```

Listing 2 PaletteToolbar Class The PaletteToolbar class implements the ActionListener interface and receives notification when a button is pressed.

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.border.*;

public class PaletteToolbar extends ToolPanel
  implements ActionListener
{
  protected Color color;
  protected ArrayList listeners = new ArrayList();

  private PaletteButton[] buttons =
  {
    new PaletteButton(Color.red),
    new PaletteButton(Color.cyan),
    new PaletteButton(Color.black),
    new PaletteButton(Color.orange),
    new PaletteButton(Color.green),
    new PaletteButton(Color.magenta),
    new PaletteButton(Color.gray),
    new PaletteButton(Color.pink),
    new PaletteButton(Color.blue),
    new PaletteButton(Color.yellow),
    new PaletteButton(Color.white),
    new PaletteButton(Color.lightGray)
  };

  public PaletteToolbar()
  {
    setBorder(new CompoundBorder(
      new EtchedBorder(),
      new EmptyBorder(4, 4, 4, 4)));
    setLayout(new GridLayout(4, 3, 2, 2));
    ButtonGroup group = new ButtonGroup();
    for (int i = 0; i < buttons.length; i++)
    {
      add(buttons[i]);
      group.add(buttons[i]);
      buttons[i].addActionListener(this);
    }
  }

  public void actionPerformed(ActionEvent event)
  {
    PaletteButton button = (PaletteButton)event.getSource();
    color = button.getColor();
    fireActionEvent("Select");
  }

  public Color getColor()
  {
    return color;
  }
}
```

are not members of a ButtonGroup. In every other way, however, ActionToolbar is like DrawToolbar. The ActionButton is identical to a DrawButton, except for the need to store a cursor. Like the DrawConstants, the ActionConstants define file prefixes for the icons, but we don't expect any files with the Cursor suffix to be loaded.

Pixel Space

One key element of an icon editor is the pixel editing space. Our implementation uses two classes in the form of a PixelView, which lets us render pixel representations in a larger visible area, and a PixelEditor, which extends the PixelView by adding editing capabilities. This arrangement supports a nice separation of responsibilities and provides additional flexibility by promoting reuse should we decide in the future that we want a lightweight view without the editing features.

The PixelView source code is shown in Listing 4. The constructor expects a BufferedImage and a cell size value. The BufferedImage is our source image, which will be drawn by mapping each pixel to a larger square in a grid. Each pixel is drawn based on the specified cell size parameter and will be either the color of the pixel or a single small dot if the pixel is transparent. We store the image and cell size as instance variables to be used in our paint method.

To make sure we can use our PixelView with larger images, probably in a scroll view, we provide a getPreferredSize() method, which does a simple calculation based on the cell size and possible insets. It's important to consider insets if we want to respond correctly when a border is used.

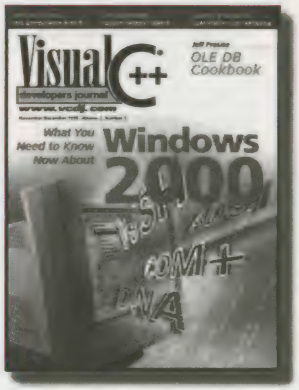
Drawing pixels is a simple matter of iterating through the BufferedImage using the getRGB() method for each pixel position. Before we draw the pixels, our paintComponent() method draws vertical and horizontal grid lines. We delegate the pixel draw-

Continued on page 56.

Become a charter subscriber NOW!

PROGRAM BETTER AND FASTER. Subscribe Now.

$19.95 12 issues annually



Name

Address

City State Zip

E-mail

O Payment Enclosed O Bill Me O Charge My: ○ Mastercard ○ VISA ○ AMEX

Card No. Exp. Date

Signature

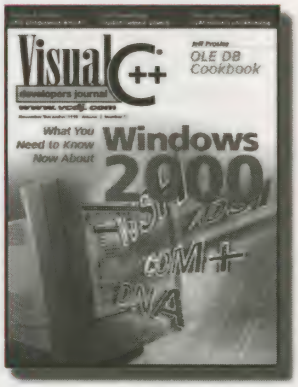
One year is 12 monthly issues. Please allow 4–6 weeks for delivery of first issue. International subscriptions must be payable in U.S. dollars plus postage. Canada and Mexico add $18/year, all other countries add $44/year for airmail delivery. Canadian GST included.

4903

Become a charter subscriber NOW!

PROGRAM BETTER AND FASTER. Subscribe Now.

$19.95 12 issues annually



Name

Address

City State Zip

E-mail

O Payment Enclosed O Bill Me O Charge My: ○ Mastercard ○ VISA ○ AMEX

Card No. Exp. Date

Signature

One year is 12 monthly issues. Please allow 4–6 weeks for delivery of first issue. International subscriptions must be payable in U.S. dollars plus postage. Canada and Mexico add $18/year, all other countries add $44/year for airmail delivery. Canadian GST included.

4903

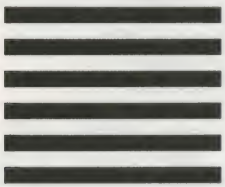
NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Visual C++
Developers Journal
P.O. Box 59289
Boulder, CO 80323-9289

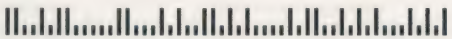
NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Visual C++
Developers Journal
P.O. Box 59289
Boulder, CO 80323-9289



Code Faster, CODE Smarter

Your Technical Guide to Faster, Better VC++ Code

You're an experienced, savvy Visual C++ professional, and *VCDJ* is your indispensable tool for faster, smarter code!

There's no better independent guide to hardcore, professional Visual C++ programming. You can't afford to miss one issue, or even one article, of *VCDJ's* practical advice for corporate development. You'll find timesaving tips and proven techniques for building server-side components, client executables, Web apps, or anything Visual C++ can create. Each issue will bring you hands-on, code-intensive articles that help you:

- Turbocharge your Visual C++ apps' performance
- Choose the right features and functions to balance speed and functionality
- Expose the right COM interfaces in the right way
- Master the latest Visual C++ techniques and tools
- Control memory allocation like a pro
- Take advantage of new programming paradigms
- Access data fast
- Harness the power of XML

Visual C++ Developers Journal

Don't miss a single issue.

Order online

www.vcdj.com

Order by phone: 1-800-848-5523

Outside the U.S. : 1-650-833-7100

VCDJ is another high-quality, how-to publication from Fawcette Technical Publications

VISUAL C++ IS A REGISTERED TRADEMARK OF MICROSOFT CORP. ALL OTHER PRODUCT NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS.

Listing 3 DrawButton Class We use an ImageIcon to store both the Icon and Cursor images in DrawButton. The preferred size and icons are set up in the constructor, along with a command name and a reference to the cursor for later use.

```
import java.io.*;
import java.awt.*;
import javax.swing.*;

public class DrawButton extends JToggleButton
  implements DrawConstants
{
  protected String command;
  protected Cursor cursor;

  public DrawButton(String name, Point point)
  {
    String iconfile = "icons/" + name + "Icon.GIF";
    String cursfile = "icons/" + name + "Cursor.GIF";
    ImageIcon icon = new ImageIcon(iconfile);
    setIcon(icon);
    setPreferredSize(new Dimension(22, 22));
    setMaximumSize(new Dimension(22, 22));
    command = name;
    if ((new File(cursfile)).exists())
    {
      ImageIcon curs = new ImageIcon(cursfile);
      cursor = Toolkit.getDefaultToolkit().createCustomCursor(
        curs.getImage(), point, command);
    }
    else
      cursor = null;
    setFocusPainted(false);
  }

  public boolean isFocusTraversable()
  {
    return false;
  }

  public boolean isDefaultButton()
  {
    return false;
  }

  public String getCommand()
  {
    return command;
  }

  public Cursor getToolCursor()
  {
    if (cursor == null)
      return Cursor.getDefaultCursor();
    return cursor;
  }
}
```

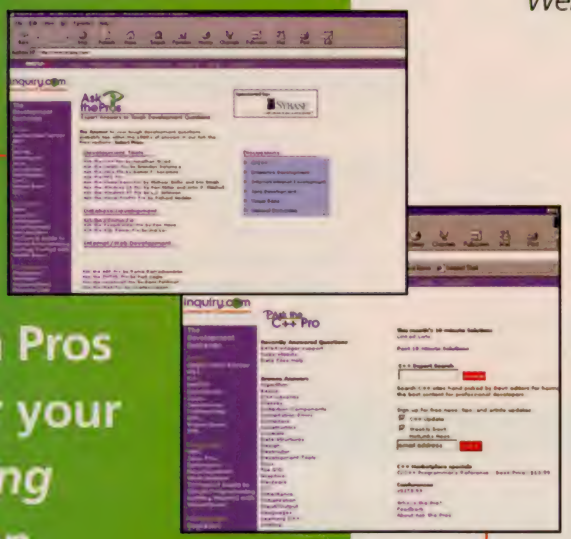
Listing 4 PixelView Class The constructor in this class expects a BufferedImage and a cell size value.

```
import java.awt.*;
import java.awt.image.*;
import javax.swing.*;

public class PixelView extends JPanel
{
  protected int cellSize;
  protected BufferedImage image;

  public PixelView(BufferedImage image, int cellSize)
  {
    this.image = image;
    this.cellSize = cellSize;
  }

  public Dimension getPreferredSize()
  {
    Insets insets = getInsets();
    int width = image.getWidth();
    int height = image.getHeight();
    return new Dimension(
      insets.left + insets.right + width * cellSize + 1,
      insets.top + insets.bottom + height * cellSize + 1);
  }

  public void paintComponent(Graphics g)
  {
    int w = getSize().width;
    int h = getSize().height;
    g.setColor(getBackground());
    g.fillRect(0, 0, w, h);

    Insets insets = getInsets();
    int width = image.getWidth();
    int height = image.getHeight();

    // Draw grid
    g.setColor(Color.lightGray);
    for (int x = 0; x <= width; x++)
    {
      int offset = insets.left + x * cellSize;
      g.drawLine(offset, insets.top,
        offset, insets.top + height * cellSize);
    }
    for (int y = 0; y <= height; y++)
    {
      int offset = insets.top + y * cellSize;
      g.drawLine(insets.left, offset,
        insets.left + width * cellSize, offset);
    }
    // Draw cells
    for (int x = 0; x < width; x++)
    {
      for (int y = 0; y < height; y++)
      {
        drawCell(g, insets, x, y, image.getRGB(x, y));
      }
    }
  }

  protected void drawCell(Graphics g,
    Insets insets, int x, int y, int rgb)
  {
    int alpha = (rgb >> 24) & 0xff;
    if (alpha == 0)
    {
      g.setColor(getBackground());
      g.fillRect(
        insets.left + x * cellSize + 1,
        insets.top + y * cellSize + 1,
        cellSize - 1, cellSize - 1);
      g.setColor(Color.black);
      g.drawLine(
        insets.left + x * cellSize + (cellSize / 2),
        insets.top + y * cellSize + (cellSize / 2),
        insets.left + x * cellSize + (cellSize / 2),
        insets.top + y * cellSize + (cellSize / 2));
    }
    else
    {
      g.setColor(new Color(rgb));
      g.fillRect(
        insets.left + x * cellSize + 1,
        insets.top + y * cellSize + 1,
        cellSize - 1, cellSize - 1);
    }
  }
}
```

Get Expert IT Development Answers Fast—*Free!*

When you need smart answers to tough development questions, go to www.inquiry.com

Search for answers to your programming questions, or ask your own question of one of our 18 programming experts in application, Web, or database development.



Inquiry.com Pros will answer your *programming* questions on multiple subjects including:

- Visual Basic
- JavaScript
- SQL Server
- PowerBuilder
- C++
- Dynamic HTML
- *And Many More!*

- **Comprehensive**
  As part of The Development Exchange network of technical sites, you also have access to the largest database of independent product reviews, tips, and information about more than 5,000 development tools.
- **Efficient**
  Nothing comes close to the efficiency and scope of inquiry.com's timely answers to real-world development problems.
- **Technical**
  Our editors and "Ask the Pro" professionals provide experience and expertise insights on techniques and tools you can use to help you get your job done better and faster.

Click to the pro of your choice and get the answers you need!

inquiry.com
answers for IT professionals

Fawcette Technical Publications
209 Hamilton Avenue, Palo Alto, CA 94301
(650)833-7100
www.devx.com

Inquiry.com, DevX, and The Development Exchange are trademarks of Fawcette Technical Publications

Listing 5 PixelEditor Class This class extends PixelView and adds some useful behaviors.

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
import javax.swing.*;
import javax.swing.border.*;

public class PixelEditor extends PixelView
  implements DrawConstants, ActionConstants,
    MouseListener, MouseMotionListener, ActionListener
{
  protected String activeTool = PENCIL;
  protected Color color = Color.blue;
  protected BufferedImage move;
  protected Rectangle drag;
  protected JLabel preview;
  protected ImageOps imageOps;
  protected JFileChooser fileChooser;

  public PixelEditor(BufferedImage image,
    int cellSize, JLabel preview)
  {
    super(image, cellSize);
    this.preview = preview;
    imageOps = new ImageOps();
    addMouseListener(this);
    setBorder(new EmptyBorder(4, 4, 4, 4));
    fileChooser = new JFileChooser();
    fileChooser.addChoosableFileFilter(new GIFFileFilter());
    fileChooser.setCurrentDirectory(new File("."));
  }

  protected void refresh()
  {
    repaint();
    preview.repaint();
  }

  public void paintComponent(Graphics g)
  {
    super.paintComponent(g);
    if (drag != null)
    {
      drawSelection(g, getInsets(), activeTool);
    }
  }

  protected void drawSelection(Graphics g,
    Insets insets, String selected)
  {
    g.setColor(color);
    int halfCell = cellSize / 2;
    int corner = cellSize * 3;
    Rectangle scaled = new Rectangle(
      insets.left + drag.x * cellSize + halfCell,
      insets.top + drag.y * cellSize + halfCell,
      drag.width * cellSize, drag.height * cellSize);
    if (selected == LINE)
    {
      g.drawLine(scaled.x, scaled.y,
        scaled.x + scaled.width,
        scaled.y + scaled.height);
    }
    if (selected == FILL_OVAL)
    {
      g.fillOval(scaled.x, scaled.y,
        scaled.width, scaled.height);
    }
    if (selected == OVAL ||
        selected == FILL_OVAL)
    {
      g.drawOval(scaled.x, scaled.y,
        scaled.width, scaled.height);
    }
    if (selected == FILL_RECT)
    {
      g.fillRect(scaled.x, scaled.y,
        scaled.width, scaled.height);
    }
    if (selected == RECT ||
        selected == FILL_RECT)
    {
      g.drawRect(scaled.x, scaled.y,
        scaled.width, scaled.height);
    }
    if (selected == FILL_ROUND_RECT)
    {
      g.fillRoundRect(scaled.x, scaled.y,
        scaled.width, scaled.height, corner, corner);
    }
    if (selected == ROUND_RECT ||
        selected == FILL_ROUND_RECT)
    {
      g.drawRoundRect(scaled.x, scaled.y,
        scaled.width, scaled.height, corner, corner);
    }
  }

  public void mouseClicked(MouseEvent event) {}
  public void mouseEntered(MouseEvent event) {}
  public void mouseExited(MouseEvent event) {}

  public void mouseReleased(MouseEvent event)
  {
    removeMouseMotionListener(this);
    if (drag != null)
    {
      Graphics2D g = (Graphics2D)image.getGraphics();
      g.setColor(color);
      g.setRenderingHint(
        RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);
      if (activeTool == LINE)
      {
        g.drawLine(drag.x, drag.y,
          drag.x + drag.width, drag.y + drag.height);
      }
      if (activeTool == FILL_OVAL)
      {
        g.fillOval(drag.x, drag.y, drag.width, drag.height);
      }
      if (activeTool == OVAL ||
          activeTool == FILL_OVAL)
      {
        g.drawOval(drag.x, drag.y, drag.width, drag.height);
      }
      if (activeTool == FILL_RECT)
      {
        g.fillRect(drag.x, drag.y, drag.width, drag.height);
      }
      if (activeTool == RECT ||
          activeTool == FILL_RECT)
      {
        g.drawRect(drag.x, drag.y, drag.width, drag.height);
      }
      if (activeTool == FILL_ROUND_RECT)
      {
        g.fillRoundRect(drag.x, drag.y, drag.width,
          drag.height, 3, 3);
      }
      if (activeTool == ROUND_RECT ||
          activeTool == FILL_ROUND_RECT)
      {
        g.drawRoundRect(drag.x, drag.y, drag.width,
          drag.height, 3, 3);
      }
    }
```

Continued on page 54.

www.java-zone.com

100% Pure Development Solutions. Delivered Daily Online.

It's all here on the DevX.com Java Zone

Jump on today and get immediate access to the most comprehensive source of expert advice, information, code and essential resources to help you develop serious Java applications. You'll get:

- **Technical articles and code** from *Java™ Pro*, *Enterprise Development*, and *Technical Guide to Visual Programming*
- **Daily programming tips**—plus a library of thousands more
- **Java tools**—new applications and applets featured weekly
- **Product reviews**—hundreds of reviews updated weekly
- **Forums**—dedicated to Java development
- **Search products**—from our database of over 5,000 programming tools
- **40% off Books of the Week**—plus editors' picks and reviews of top programming titles
- **Order and download** the top software titles for Java development

And that's just the beginning.

DEV·X The Development Exchange

Explore the rest yourself at
www.java-zone.com

Java-Zone is a member of the family of Windows and interactive development sites located on The Development Exchange at www.devx.com. Java is a registered trademark of Sun Microsystems. The Development Exchange and DevX are trademarks of Fawcette Technical Publications. Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500. Phone: 650.833.7100; Fax: 650.853.0230.

Continued from page 52.

```
      move = null;
      drag = null;
      refresh();
    }

    public void mousePressed(MouseEvent event)
    {
      Insets insets = getInsets();
      int width = image.getWidth();
      int height = image.getHeight();
      int x = (event.getX() - insets.left) / width;
      int y = (event.getY() - insets.top) / height;
      if (activeTool.equals(PENCIL))
      {
        image.setRGB(x, y, color.getRGB());
      }
      if (activeTool.equals(ERASER))
      {
        image.setRGB(x, y, 0);
      }
      if (activeTool.equals(DROPPER))
      {
        color = new Color(image.getRGB(x, y));
      }
      if (activeTool.equals(PAINT))
      {
        imageOps.floodFill(image, x, y,
          image.getRGB(x, y), color.getRGB());
      }
      if (activeTool.equals(LINE) ||
          activeTool.equals(OVAL) ||
          activeTool.equals(FILL_OVAL) ||
          activeTool.equals(RECT) ||
          activeTool.equals(FILL_RECT) ||
          activeTool.equals(ROUND_RECT) ||
          activeTool.equals(FILL_ROUND_RECT))
      {
        int xa = (event.getX() - insets.left) / cellSize;
        int ya = (event.getY() - insets.top) / cellSize;
        drag = new Rectangle(xa, ya, 0, 0);
        addMouseMotionListener(this);
      }
      if (activeTool.equals(HAND))
      {
        int w = image.getWidth();
        int h = image.getHeight();
        move = new BufferedImage(
          w, h, BufferedImage.TYPE_INT_ARGB);
        Graphics g = move.getGraphics();
        g.drawImage(image, 0, 0, this);
        int xa = (event.getX() - insets.left) / cellSize;
        int ya = (event.getY() - insets.top) / cellSize;
        drag = new Rectangle(xa, ya, 0, 0);
        addMouseMotionListener(this);
      }
      refresh();
    }

    public void mouseMoved(MouseEvent event) {}
    public void mouseDragged(MouseEvent event)
    {
      Insets insets = getInsets();
      if (move != null)
      {
        int x = (event.getX() - insets.left) /
          cellSize - drag.x;
        int y = (event.getY() - insets.top) /
          cellSize - drag.y;
        int w = image.getWidth();
        int h = image.getHeight();
        image = new BufferedImage(
          w, h, BufferedImage.TYPE_INT_ARGB);
        Graphics g = image.getGraphics();
        g.drawImage(move, x, y, this);
        refresh();
      }
      if (drag != null)
      {
        int x = (event.getX() - insets.left) / cellSize;
        int y = (event.getY() - insets.top) / cellSize;
        drag.setSize(x - drag.x, y - drag.y);
        refresh();
      }
    }

    public void actionPerformed(ActionEvent event)
    {
      Object source = event.getSource();
      if (source instanceof PaletteToolbar)
      {
        PaletteToolbar palette = (PaletteToolbar)source;
        color = palette.getColor();
      }
      if (source instanceof DrawToolbar)
      {
        DrawToolbar tools = (DrawToolbar)source;
        activeTool = event.getActionCommand();
        setCursor(tools.getToolCursor());
      }
      if (source instanceof ActionToolbar)
      {
        ActionToolbar toolbar = (ActionToolbar)source;
        String command = event.getActionCommand();
        if (command.equals(MOVE_NORTH_WEST))
          image = imageOps.moveImage(image, -1, -1);
        if (command.equals(MOVE_NORTH))
          image = imageOps.moveImage(image, 0, -1);
        if (command.equals(MOVE_NORTH_EAST))
          image = imageOps.moveImage(image, 1, -1);
        if (command.equals(MOVE_WEST))
          image = imageOps.moveImage(image, -1, 0);
        if (command.equals(MOVE_EAST))
          image = imageOps.moveImage(image, 1, 0);
        if (command.equals(MOVE_SOUTH_WEST))
          image = imageOps.moveImage(image, -1, 1);
        if (command.equals(MOVE_SOUTH))
          image = imageOps.moveImage(image, 0, 1);
        if (command.equals(MOVE_SOUTH_EAST))
          image = imageOps.moveImage(image, 1, 1);
        if (command.equals(FLIP_VERT))
          image = imageOps.flipImage(image, ImageOps.VERTICAL);
        if (command.equals(FLIP_HORZ))
          image = imageOps.flipImage(
            image, ImageOps.HORIZONTAL);
        if (command.equals(ROTATE_RIGHT))
          image = imageOps.rotateImage(image, ImageOps.RIGHT);
        if (command.equals(ROTATE_LEFT))
          image = imageOps.rotateImage(image, ImageOps.LEFT);

        if (command.equals(NEW_FILE))
          newFile();
        if (command.equals(OPEN_FILE))
          openFile();
        if (command.equals(SAVE_FILE))
          saveFile();

        refresh();
      }
    }

    protected void newFile()
    {
      int w = image.getWidth();
      int h = image.getHeight();
      image = new BufferedImage(
        w, h, BufferedImage.TYPE_INT_ARGB);
      preview.setIcon(new ImageIcon(image));
      refresh();
```

Setting the Pace for Tomorrow's Internet—Today!

Register before **February 4!** and **Save $225**

Plan to Attend the Largest Gathering of APACHE users!

APACHECON 2000

Conference: March 8-10, 2000
Exhibition: March 9-10, 2000
Caribe Royale Resort Suites & Villas
Orlando, Florida

Apache's Support Keeps Growing

More than half of Web sites on the Internet choose to run Apache. Apache's reliability, flexibility and speed offer Web site administrators unparalleled customized solutions.

Join more than 1,000 of your peers at the largest gathering of Apache users. The language's most respected technical experts, sought-after gurus and advanced users are converging at **ApacheCon 2000** for one reason—to explore how you can get the most out of Apache. Immerse yourself in 3 days of Apache intelligence. Learn directly from those who helped build the software supporting more than half of the web sites in the Internet today.

Event Highlights

- Customize your program. Select from 40 sessions with over 70+ hours of instruction.
- Night School Sessions—select from 8 in-depth topics on Wednesday or Thursday night.
- Get the latest insights and useable advice on the latest Apache projects and enhancements driving the Open Source revolution.
- Topics Include: Apache and eCommerce, Java, XML, PHP, Performance, Perl, Security, Windows.
- Designed for Apache and Open Source enthusiasts.
- Two-day Interactive Exhibition on Thursday and Friday.
- The only Apache event to be fully supported by the Apache Software Foundation.

Who Should Attend?

Anyone seeking to better understand Open Source and Apache software features and benefits. It is designed for:

- Open source software developers
- Apache software developers
- Web site administrators
- Technical managers responsible for running Web sites

What Makes This Event So Valuable to You?

- You'll learn about new Apache projects and enhancements.
- The tips and techniques you'll learn will help you do your job better.
- Discover new applications being developed today that you will need tomorrow.
- Sessions are designed for the users at all levels.
- You will have the opportunity to network with fellow software developers as well as recognized Apache experts.

Attention Project Managers—Special Team Discount

Send three or more of your team members and each will save $250 off the regular three-day rate.

Explore Today's Hottest Topics

ApacheCon 2000 is the only Apache event to be fully supported by the Apache Software Foundation. Apache founders and leading contributors have designed the technical program which includes 40 sessions with over 70+ hours of instruction. Each session is designed to be immediately useful and is sure to attract serious developers and IT management. Topics covered include:

- Getting Started with mod_perl
- Design Patterns in Web Programming
- XML Publishing Fundamentals
- Comanche, a GUI for Apache
- Building an XML Web Application
- Secure Financial Transactions with Open Source
- Apache and LPAD
- HTTP and Apache
- Apache: A Business Server
- Performance Tuning Java Code in Jakarta
- Transparent Content Negotiation
- Apache on Windows
- PHP: Hacker's Paradise
- Practical XSLT
- SSL Tutorial
- Using C++ for Apache Modules
- XML Parsing

Presented by The Apache Software Foundation

Produced by CAMELOT COMMUNICATIONS

Corporate Sponsors: IBM, Sun microsystems, Collab.Net

Media Sponsors: Apacheweek, InfoWorld, NetCraft, JavaPro, XML, Visual Basic, Enterprise, DevX, Dr. Dobb's, EarthWeb

For the latest updates and to register, visit: http://www.apachecon.com

Continued from page 48.

ing to a method called drawPixel(), which decides whether to draw a filled-in square the color of the pixel or a small dot on a flat background if the alpha value is zero.

The PixelEditor, shown in Listing 5, extends the PixelView class and adds useful behaviors. Some of these are context-dependent and based on the active tool, driven by selections in the DrawToolbar. Whenever tools are selected, we receive an action event and, after determining which toolbar the event came from, either set the context or take immediate action. In the case of color selections or drawing tools, we set the color or the active tool. Drawing tools, as you may recall, also set the cursor.

The PixelEditor class processes mouse events. The mousePressed() events are triggers for drawing operations, such as drawing or erasing a single pixel or setting an anchor for an operation that relies on the mouse release position, such as drawing a line, oval, or rectangle. The first cases are easy and effectively complete by the time the mouse is released. Other cases involve mouse dragging and drawing a rubber band-style tracking shape.

The paintComponent() method calls the superclass to draw the grid and calls the drawSelection() method if the drag variable is not null. We always make sure it gets set to null intentionally when the mouse is released and the drawing complete. The drawSelection() method calculates the drawing rectangle and draws the outline of the selection based on the color and the drawing tool that was selected.

The mouseDragged() events handle selection and movement operations. Selection is simply a matter of moving the non-anchored corner of our selection and repainting. Movement requires making a copy of the image and drawing into it at the new position. We handle this, in part, with the mousePressed() event to copy the original image to be dragged, and, in part, with the mouseDragged event to position the image at the new location. You can see this behavior when you use the hand icon to drag the image around.

After the mouse is released, a drawing operation need only draw the line, oval, or rectangle outline, and possibly filled region, into the image buffer and the repaint takes care of the rest. While the selection drawing is done on the larger PixelView canvas, releas-

Listing 6 ImageOps Class All of these operations may be used in other contexts.

```
import java.awt.*;
import java.awt.image.*;
import javax.swing.*;

public class ImageOps implements SwingConstants
{
  protected BufferedImage moveImage(
    BufferedImage image, int x, int y)
  {
    int w = image.getWidth();
    int h = image.getHeight();
    BufferedImage buffer = new BufferedImage(
      w, h, BufferedImage.TYPE_INT_ARGB);
    Graphics gc = buffer.getGraphics();
    gc.drawImage(image, 0, 0, null);
    image = new BufferedImage(
      w, h, BufferedImage.TYPE_INT_ARGB);
    Graphics g = image.getGraphics();
    g.drawImage(buffer, x, y, null);
    return image;
  }

  protected BufferedImage flipImage(
    BufferedImage image, int direction)
  {
    int w = image.getWidth();
    int h = image.getHeight();
    BufferedImage buffer = new BufferedImage(
      w, h, BufferedImage.TYPE_INT_ARGB);
    Graphics gc = buffer.getGraphics();
    if (direction == HORIZONTAL)
      gc.drawImage(image, w, 0, 0, h, 0, 0, w, h, null);
    else
      gc.drawImage(image, 0, h, w, 0, 0, 0, w, h, null);
    image = new BufferedImage(
      w, h, BufferedImage.TYPE_INT_ARGB);
    Graphics g = image.getGraphics();
    g.drawImage(buffer, 0, 0, null);
    return image;
  }

  protected BufferedImage rotateImage(
    BufferedImage image, int direction)
  {
    int w = image.getWidth();
    int h = image.getHeight();
    BufferedImage buffer = new BufferedImage(
      w, h, BufferedImage.TYPE_INT_ARGB);
    Graphics2D gc = (Graphics2D)buffer.getGraphics();
    if (direction == RIGHT)
    {
      gc.translate(w, 0);
      gc.rotate(Math.PI / 2);
    }
    if (direction == LEFT)
    {
      gc.translate(0, h);
      gc.rotate(-(Math.PI / 2));
    }
    gc.drawImage(image, 0, 0, null);

    image = new BufferedImage(
      w, h, BufferedImage.TYPE_INT_ARGB);
    Graphics g = image.getGraphics();
    g.drawImage(buffer, 0, 0, null);
    return image;
  }

  protected void floodFill(BufferedImage buffer,
    int x, int y, int oldColor, int newColor)
  {
    int w = buffer.getWidth();
    int h = buffer.getHeight();
    if (y < 0 || y >= h) return;
    if (x < 0 || x >= w) return;
    int pos = x;
    while (buffer.getRGB(pos, y) == oldColor)
    {
      buffer.setRGB(pos, y, newColor);
      if (y > 0 && buffer.getRGB(pos, y - 1) == oldColor)
        floodFill(buffer, pos, y - 1, oldColor, newColor);
      if ((y + 1) < h && buffer.getRGB(pos, y + 1) == oldColor)
        floodFill(buffer, pos, y + 1, oldColor, newColor);
      pos--;
      if (pos < 0) break;
    }
    pos = x + 1;
    if (pos >= w) return;
    while (buffer.getRGB(pos, y) == oldColor)
    {
      buffer.setRGB(pos, y, newColor);
      if (y > 0 && buffer.getRGB(pos, y - 1) == oldColor)
        floodFill(buffer, pos, y - 1, oldColor, newColor);
      if ((y + 1) < h && buffer.getRGB(pos, y + 1) == oldColor)
        floodFill(buffer, pos, y + 1, oldColor, newColor);
      pos++;
      if (pos >= w) break;
    }
  }
}
```

Try Premier Club RISK FREE for 30 days

DEV·X
The Development Exchange

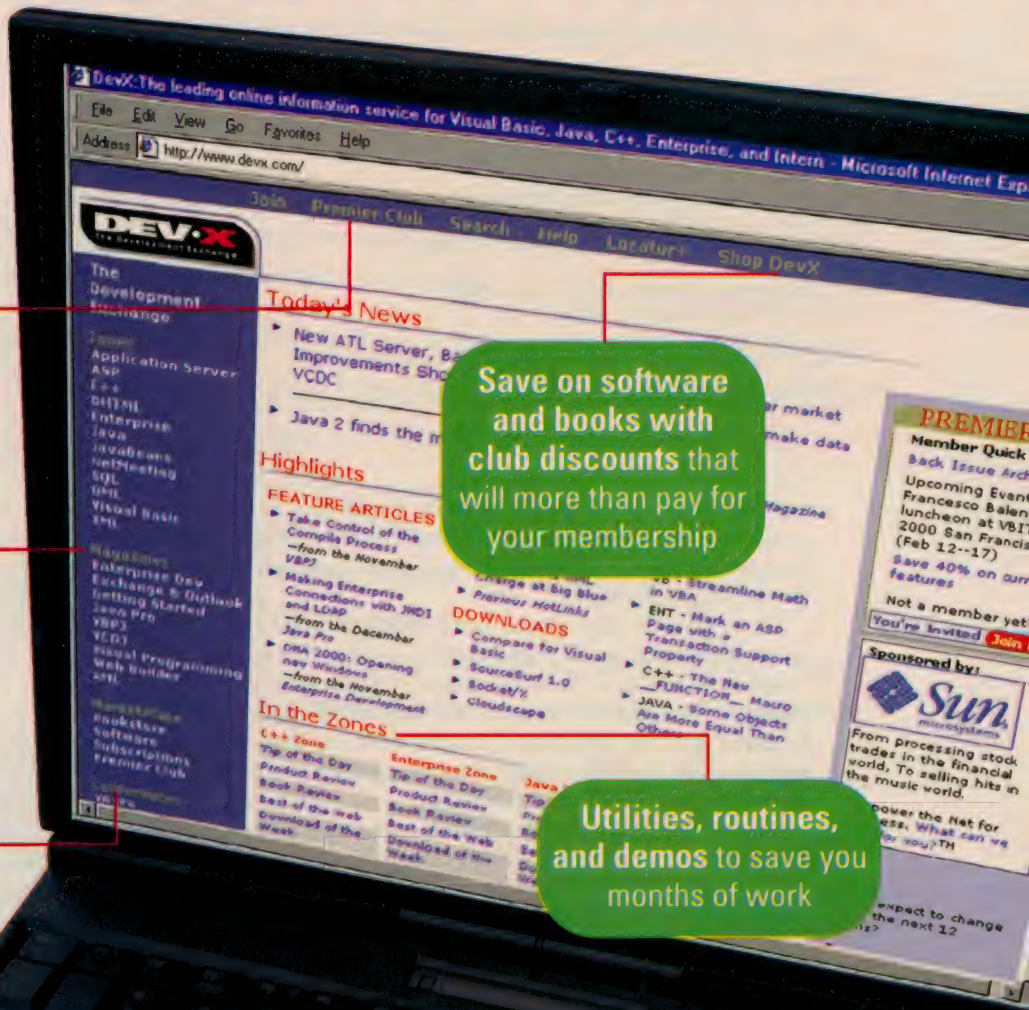
Discover the unlimited resources and exclusive benefits thousands of professional developers rely on everyday at:

www.devx.com

Full access to our searchable archive of articles and code from FTP print magazines

Gigabytes of free code

Live member-only events with programming experts


DevX: The leading online information service for Visual Basic, Java, C++, Enterprise, and Intern - Microsoft Internet Exp
File Edit View Go Favorites Help
Address http://www.devx.com/
Join Premier Club Search Help Locator Shop DevX
Today's News
Highlights
FEATURE ARTICLES
DOWNLOADS
In the Zones
Sponsored by:
Sun

Save on software and books with club discounts that will more than pay for your membership

Utilities, routines, and demos to save you months of work

Join now and your first 30 days are FREE

www.devx.com/freepremier and enter access code 5008PCE

Or Call 1-800-848-5523

Just purchase your Premier Club membership and the first 30 days are yours free*.
If at any time during your 30-day trial period you are not completely satisfied with your Premier Club membership, you can cancel your membership and receive a full refund.

Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301
To process your order offline call 1-800 -848-5523 or 650-833-7100 (outside U.S.) 8:30-5:30 Pacific time

DEV·X The Development Exchange

THE 30-DAY FREE PREMIER CLUB MEMBERSHIP TRIAL MUST BE USED WITHIN ONE MONTH OF SIGNING UP AND IS AVAILABLE TO ALL NEW MEMBERS OF DEVX.COM. A MAJOR CREDIT CARD IS REQUIRED IN ORDER TO SIGN UP. TO AVOID BEING CHARGED YOUR FIRST MONTHLY FEE, SIMPLY CANCEL YOUR MEMBERSHIP BEFORE THE TRIAL PERIOD ENDS. Offer expires January 2000. DevX is a registered trademark of Fawcette Technical Publications. All other names are trademarks or service marks of their respective holders

Listing 7 JIconEditor Class The JIconEditor class implements the editor in a JPanel for convenience.

```
import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
import javax.swing.*;
import javax.swing.border.*;

public class JIconEditor extends JPanel
{
  public JIconEditor()
  {
    BufferedImage image = readBufferedImage(
      "icons/Computer.gif");

    setLayout(new BorderLayout());

    JPanel west = new JPanel(new BorderLayout());
    JPanel east = new JPanel(new BorderLayout());

    ToolPanel action = new ActionToolbar();
    ToolPanel draw = new DrawToolbar();
    JPanel westPanel = new JPanel(new BorderLayout());
    westPanel.add(BorderLayout.NORTH, action);
    westPanel.add(BorderLayout.SOUTH, draw);

    west.add(BorderLayout.NORTH, westPanel);
    add(BorderLayout.WEST, west);

    JLabel preview = new JLabel(new ImageIcon(image));
    preview.setPreferredSize(new Dimension(26, 26));
    preview.setBorder(BorderFactory.createEtchedBorder());

    ToolPanel palette = new PaletteToolbar();

    JPanel eastPanel = new JPanel(new BorderLayout());
    eastPanel.add(BorderLayout.NORTH, preview);
    eastPanel.add(BorderLayout.SOUTH, palette);

    east.add(BorderLayout.NORTH, eastPanel);
    add(BorderLayout.EAST, east);

    PixelEditor editor = new PixelEditor(image, 16, preview);
    add(BorderLayout.CENTER, new JScrollPane(editor));

    palette.addActionListener(editor);
    draw.addActionListener(editor);
    action.addActionListener(editor);
  }

  public BufferedImage readBufferedImage(String filename)
  {
    ImageIcon icon = new ImageIcon(filename);
    Image image = icon.getImage();
    BufferedImage buffer = new
      BufferedImage(image.getWidth(null),
      image.getHeight(null),BufferedImage.TYPE_INT_ARGB);
    Graphics g = buffer.getGraphics();
    g.drawImage(image, 0, 0, null);
    return buffer;
  }
}
```

ing the mouse causes painting to be transferred to the original image and the repaint causes the larger view to be updated to reflect the changes. We use the Java 2D antialiasing hints to make the drawing smoother.

The ActionToolbar events use a subclass called ImageOps to move, flip, rotate, or fill portions of an image. This helps keep the coupling looser and allows us to reuse image operations in other contexts. The only tricky operation is the floodFill algorithm, which required a lot of tweaking. I started with my own version, researched it on the Web to find something better, and found myself repairing the best version I found.

The floodFill algorithm is recursive and works by moving out from a center point in every direction. The center point moves around through recursive calls and works its way around corners to fill everything that touches the original pixel in the same color, except for diagonals. It may not be optimally efficient on very large images, but it works reliably in all cases and acts plenty fast for an icon editor.

All the operations in the ImageOps class (see Listing 6) operate on a BufferedImage and may be used in other contexts, making the code as reusable as possible. The PixelEditor is more tightly coupled to the operations it performs and could be improved by developing a model that delegates operations to other classes. I found the compromise between the number of classes, overall organization, and loose coupling reasonable at this level, though I might use keystroke mappings and Action objects in a future version.

Top of the Class

Listing 7 shows the JIconEditor class, which ties all this together, implementing the editor in a JPanel for convenience. The ActionToolbar and DrawToolbar are placed on the left and the PaletteToolbar on the right of a PixelView area. The PixelView is in a JScrollPane in the event that the image is too large to fit in the viewing area. With the real work delegated to subcomponents, there's nothing more than a constructor that positions the elements in a BorderLayout.

You'll find a JIconEditorTest class online when you pull the source code down. The main method simply creates a JFrame and puts a JIconEditor in the center. I found loading time to be slightly longer than I like because so many small images were loaded for the toolbar icons and cursors. This setup could be optimized by putting all the images together in a single image/grid and reconstructing the individual icons in memory.

JIconEditor is a useful tool for editing small images. The toolset is relatively comprehensive and provides considerable power to a user. This component is more of a small application than those we've visited in past installments, yet it provides a drop-in set of functionality that can help enhance your user experience. Don't drop an icon editor into an end-user application that doesn't need it, but if the moment is right, you know where to look for the tools. ■

Claude Duguay is the Software Engineering Manager at Atrieva Corp. in Seattle. He has written for several programming magazines and has been programming for nearly two decades. Reach him at claude@atrieva.com.

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0002 Download all of the code for this issue of *Java Pro*. Free online registration is required.

JP0002VC DevX Premier Club members download the code for this article.

JP0002VC_T DevX Premier Club members can read this feature online.

Not a member yet? Go to www.DevX.com to sign up for the free Registered Level or to subscribe to the Premier Club.

HANDS-ON TECHNICAL SOLUTIONS

UNLOCK the POWER

Introducing *Exchange & Outlook* Magazine,

the only independent magazine dedicated to IT professionals, administrators, managers and corporate developers using Microsoft Exchange and Microsoft Outlook technologies.

Exchange & Outlook is written by technical professionals who understand the everyday challenges you face.

Packed with hands-on, in-depth technical information, each page of *Exchange & Outlook* magazine contains proven strategies and no-nonsense tips, tricks and techniques, *Exchange & Outlook* delivers exactly what you need, including:

- How to plan for, deploy, and leverage Exchange 2000
- Critical independent evaluations of new products
- Essential, real-world administration tips
- Code you can use today to build collaborative applications
- Proven deployment strategies for custom applications
- Customizing Outlook 97/98 and Outlook 2000
- Complete groupware projects and strategies

SPECIAL CHARTER SUBSCRIPTION OFFER: ONLY $19.99

Subscribe online at:

http://exchange.devx.com/in.asp

Or order by phone, call: **1-800-848-5523**

Outside the U.S., call: 1-650-833-7100

FTP FAWCETTE TECHNICAL PUBLICATIONS

209 Hamilton Ave.
Palo Alto, CA 94301
http://exchange.devx.com

Windows, Microsoft, and Outlook are registered trademarks of Microsoft Corp. All other trademarks are the properties of their respective owners.

ImpureThoughts

Getting Sparse

Sparse matrices turn up in linear programming applications; these techniques will help you make the best of them

BY RICK GREHAN

On my way to explaining the Jama matrix algebra package in January's column, I took a peek at the structure of Java multidimensional arrays. Now it's time to examine a particular sort of 2D array that occurs frequently enough in linear programming applications that it warrants specialized storage techniques. That "particular sort" of 2D array is known as a sparse matrix.

The name says it all. A sparse matrix is sparsely populated. Most of its elements are zero. That being the case, if the matrix is large and we use a standard 2D array to store it, a lot of memory is used to store—well, nothing.

Of course, "large" is a relative term—especially in the computer world (where, a decade ago, a 100MB hard disk was regarded as "large"). Some large systems of linear equations can translate to a matrix with thousands of rows and columns. If such a matrix is sparse, the programmer should investigate ways to store that matrix other than a straightforward implementation as a 2D array.

It appears that we need a sparse matrix class that creates objects that, on the outside look to behave like a "normal" 2D array, but on the inside are more frugal with storage. I say "appears" because we can't override the array index operators in Java, so we have to be satisfied with accessor methods. In addition, some applications that use a sparse matrix are hampered if the matrix behaves like a normal array.

Hashtable

A first stab at managing a sparse matrix would be to create a kind of data structure that uses the row and column coordinates of the non-zero elements as a key into a database. The nonzero values would be the "payload" associated with the key. (Donald Knuth hints at this technique in Volume 1 of *The Art of Computer Programming*. We'll come back to this book later when we look at another sparse matrix data structure.)

In Java, it is easy to use the coordinates of the nonzero elements as a key into a database using the Hashtable class as the database mechanism. Listing 1 shows an example sparse matrix class implemented using a hashtable. It provides two accessor methods: get() and set(). The get() method is coded so that if a requested element isn't found in the array, it returns a 0. Furthermore, if you set() a nonzero element to zero, that element is removed from the hashtable.

This approach has its advantages and disadvantages. Perhaps the greatest advantage is that constructing it is simple. The hashtable key is simply a Long object that consists of the concatenated row and column values; the data item is the associated nonzero double value. Another advantage is that the Hashtable class provides insert and delete operations for which the coding is straightforward. The possible problems that arise from a collision (trying to insert a key/data whose key already exists in the table) is automatically handled by the Hashtable.put() method.

The not-so-obvious disadvantage is that this structure does not support fast row or column operations. To locate the next nonzero element in a row (or column), your program would have to query the hashtable repeatedly—either incrementing or decrementing the appropriate coordinate. The result, of course, would be piles and piles of "key not found" conditions until the program stumbled onto the next nonzero element. Why this might be important requires a brief detour.

Many linear algebra and linear programming operations on matrices involve algorithms that—at their heart—"walk" sequentially through a matrix row or column (or both). You'll find a classic example in the unadulterated matrix multiplication algorithm. Simply put, if you want to multiply two matrices A and B such that C=A×B, then the core formula is:

```
c[i][j] = a[i][0] * b[0][i] + a[i][1] *
  b[1][i] + ...
```

In other words, the result is the sum of the element-by-element multiplication of a row in A by a column in B. (Other algorithms for matrix multiplication reduce the number of multiplications and additions required. I've chosen the "classic" approach for illustration.)

If the matrices being multiplied are sparse and we use the Hashtable class approach for storage, it's easy to see that the algorithm will waste time wading through zeroes. Nevertheless, for applications in which the sparse matrix elements were accessed randomly, a hashtable approach is an easy-to-assemble solution. If the Hashtable class methods are implemented in na-

tive code, reading and writing sparse matrix elements should be acceptably quick.

Knuth's Web

One solution to our sparse matrix problem can be found if we return to *The Art of Computer Programming*, where Don Knuth describes a data structure that consists of interconnected linked lists. One set of lists wires the column elements together; another wires the row elements together.

A variation on Knuth's structure is implemented in Listing 2. (It's a variation because Knuth's lists cycle in the opposite direction from the one I've shown.) Figure 1 is a diagram of the data structure as it would appear for the simple 3×3 matrix:

```
1 0 0
0 2 0
0 0 0
```

Nonzero elements are stored in a "node," a data structure that consists of two link "pointers," a row and column coordinate, and a payload. Nonzero elements are connected along row and column, and each list is anchored by a node at the row head or column head.

The sparseMatrix class in Listing 2 provides methods for adding, removing, and retrieving elements from the matrix. The set() method is "smart" in that it properly handles the case of setting a nonzero element to zero; namely, it removes the element. I've also provided methods for locating the first and next elements in a row or column. With this structure, matrix multiplication would be less "wasteful," since the algorithm can quickly traverse rows and columns.

If the matrix is large and we use a standard 2D array to store it, a lot of memory is used to store—well, nothing

The class shown in Listing 2 simply allocates and de-allocates nodes (objects of type sparseElement) on the fly. If the values in a matrix are likely to change as the application runs, you can save time if the nodes are "pre-allocated" when the matrix is created. Nodes could be pulled from the pool as needed and returned to it when removed from the matrix. This would sidestep all the allocating and de-allocating that would otherwise take place, lightening the load on the garbage collector.

Compressed Coordinate Storage

Another technique that provides rapid access of sequential row and column members is a variation on the "coordinate storage"

Listing 1 Sparse Matrix Using a Hashtable Creating a class that uses a hashtable for sparse array access is straightforward. The row and column coordinates form the key, the nonzero value forms the data item. The hashtable handles the adding and removing of elements almost invisibly. (Note that, for speed's sake, the routine attaches no exceptions to index out-of-bounds conditions.)

```
import java.util.Hashtable;
//
// Class hashSparse
// Uses a hashtable as the storage mechanism
// for a sparse array.

class hashSparse {
  Hashtable shash;  // Hashtable
  int maxrow;       // Max row
  int maxcol;       // Max column

//
// Create a hashSparse array with max row and column
// given by mrow and mcol
public hashSparse(int mrow, int mcol)
{
  // Set max coords
  maxrow = mrow;
  maxcol = mcol;

  // Initialize the hashtable. We initialize with 100
  // elements and load factor of 0.80
  shash = new Hashtable(100, 0.80f);

  return;
}

//
// Set an item in the hashtable
// Ignore if out of bounds
// If we set an element to 0 and it exists,
// we delete the element
public void set(int row, int col, double val)
{
  long lkey; // For holding key

  // Check coords
  if((row >= maxrow) ||
    (row < 0) ||
    (col >=  maxcol) ||
    (col < 0))
    return;

  // Pack coordinates
  lkey=(long)row;
  Long key = new Long(lkey<<32 |
    (long)col);
  // If storing 0, issue a remove and return.
  if(val == (double)0.0)
  {
    shash.remove(key);
    return;
  }

  // Move val into Double object and put it in the table
  Double dval = new Double(val);
  shash.put(key,dval);
  return;
}

//
// Get an item from the hashtable
// Return 0 if out of bounds or if nothing at coordinates.
public double get(int row, int col)
{
  Double dval;
  long lkey;

  // Check coords
  if((row >= maxrow) ||
    (row < 0) ||
    (col >=  maxcol) ||
    (col < 0))
    return((double)0.0);

  // Create coordinate object and search
  lkey=(long)row;
  Long key = new Long(lkey<<32 |
    (long)col);
  dval = (Double)shash.get(key);
  if(dval == (Double)null)
    return((double)0.0);

  return(dval.doubleValue());
}
}
```

Listing 2 Sparse Matrix Using Linked Lists This code implements a "Knuth-like" sparse matrix of interconnected linked lists. Random access is not as good as with the code in Listing 1. However, this version of sparse matrix storage works well for algorithms that depend on sequential access of row and column members.

```
//
// Storage class for sparse matrix
//
public class sparseElement
{
  sparseElement right;  // Next element in row
  sparseElement down;   // Next element in column
  int row;              // Row coordinate
  int col;              // Column coordinate
  double val;           // Payload

public sparseElement()
{
  // Initialize rows and columns to -1. Useful for the
  // search algorithms
  row= -1;
  col= -1;
}
}

//
// Sparse Matrix class
// Modified "Knuth" style
//
public class sparseMatrix
{

  int rowmax;                // Max row
  int colmax;                // Max col
  sparseElement rowBase [];  // Row anchors
  sparseElement colBase [];  // Column anchors

  public sparseMatrix (int rows, int cols)
  {
    // Allocate row and column anchors
    rowBase = new sparseElement[rows];
    colBase = new sparseElement[cols];

    // Set row and column max
    rowmax = rows-1;
    colmax = cols-1;

    // Set row and col values in the anchors to -1. This
    // will help the insert algorithm
    for(int i=0; i<rows ; i++)
      rowBase[i] = new sparseElement();
    for(int i=0; i<cols ; i++)
      colBase[i] = new sparseElement();
  }

  // Add a sparseElement into the sparseMatrix. If the
  // sparseElement already exists at that coordinate,
  // this works like an update. If adding 0 and the
  // element exists, it is deleted.
  public void set (int row, int col, double val)
  {
    sparseElement elem;
    sparseElement next;
    sparseElement prev;

    // Verify in bounds -- ignore if not
    if((row < 0 ) || (row > rowmax) ||
      (col < 0) || (col > colmax))
      return;

    // Allocate and initialize element
    elem = new sparseElement();
    elem.row = row;
    elem.col = col;
    elem.val = val;

    // Wire into column. Examine column base "down" pointer.
    // If null, this is the only element. Otherwise, we have
    // to walk the list.
    if(colBase[col].down ==
      (sparseElement)null)
    { colBase[col].down = elem;
      elem.down = colBase[col];
    }
    else
    { prev = colBase[col];
      next = colBase[col].down;
      // Loop until the next element has bigger row
      // number (or -1)
      while((next.row <= row) &&
           (next.row != -1))
      { prev = next;
        next = prev.down;
      }
      // We're at the spot to insert. Verify that the element
      // doesn't already exist in the array. If it does, and
      // we are setting to zero -- delete it.
      if(prev.row==row)
      {
        if(val == (double)0.0)
          remove(prev);
        else
          prev.val=val;
        return;
      }
      // At this point, we have verified that the element
      // doesn't already exist. If we're setting to 0, exit
      if(val == (double)0.0)
        return;

      // Wire it in the column
      prev.down=elem;
      elem.down=next;
    }

    // Wire into row.
    if(rowBase[row].right ==
      (sparseElement)null)
    { rowBase[row].right = elem;
      elem.right = rowBase[row];
    }
    else
    { prev = rowBase[row];
      next = rowBase[row].right;
      // Loop as above
      while((next.col <= col ) &&
           (next.col != -1))
      { prev = next;
        next = prev.down;
      }
      // Don't have to check like above
      prev.right = elem;
      elem.right = next;
    }
    return;
  }

  // Remove sparseElement from sparseMatrix
  public void remove (sparseElement elem)
  {
    sparseElement next;

    // Loop through column, find previous element,
    // and disconnect
    next = elem.down;
    while(next.down != elem)
      next = next.down;
    next.down = elem.down;
    // Special case if we've emptied the whole column.
    if(next.col == -1)
      next.down = (sparseElement)null;

    // Loop through row
    next = elem.right;
```

Listing continued on page 64.

DEV·X
The Development Exchange
MarketPlace

TOOLS FOR PROGRAMMING PROFESSIONALS

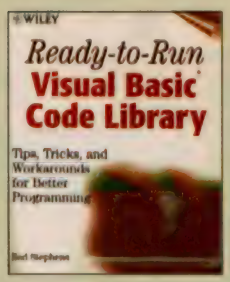
BOOKS To help you do your job better and faster

Here's just a glimpse of the books in the DevX bookstore...

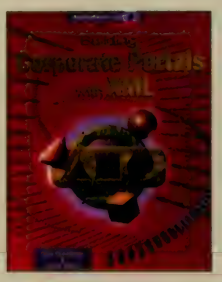
40% OFF!
HOT SPECIALS Each Week

Sams Teach Yourself Java 2 Online in Web Time
Advance your skills through book and online instruction.
Paperback w/CD
By Stephen Gilbert and Bill McCarty
Pub: $49.99
DevX Price: $39.99

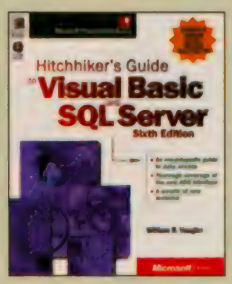
Inside ATL
Quickly build ultra-lightweight server-side objects ideally suited for multi-tier, multithreaded solutions.
Paperback w/CD
By George Shepherd and Brad King
Pub: $49.99
DevX Price: $39.99



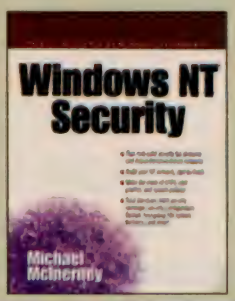
Ready-to-Run Visual Basic Code Library
An arsenal of more than 170 proven techniques and ready-to-run source code for better and more efficient programming.
Paperback w/CD
By Rod Stephens
Pub: $49.99
DevX Price: $39.99



Building Corporate Portals with XML
From planning and creating, to implementing and using XML, this is a great foundation for understanding this revolutionary technology.
Paperback
By Clive Finkelstein and Peter Aiken
Pub: $49.99
DevX Price: $39.99



Hitchhiker's Guide to VB & SQL ServerSixth Edition
The definitive book for developers who want to use VB to access Microsoft SQL Server.
Paperback w/CD
By William R. Vaughn
Pub: $49.99
DevX Price: $39.99



Windows NT Security
A solid security foundation for enterprise NT 4 and Windows 2000 networks .
Paperback
By Michael McInerney
Pub: $49.99
DevX Price: $39.99

Dynamic HTML in Action
Get the most out of the extended interactive and multimedia abilities of Dynamic HTML version 2.
Paperback w/CD
By Eric M. Schurman and William J. Pardi
Pub: $39.99
DevX Price: $31.99

Check out the latest books on these subjects:

- Access
- Active Server Pages
- ActiveX
- ADO
- C++
- Certification
- COM/DCOM
- Databases
- Dynamic HTML
- Enterprise Development
- Exchange and Outlook
- Java
- Microsoft Transaction Server
- Object Oriented Programming
- SQL Server
- Visual Basic
- Visual InterDev
- Web Development
- XML

and more!

Look for DevX.com Author Events:
Discussion, insights, views, plus autographed books!
http://news.devx.com

Order Online at
http://books.devx.com
Or call 800.848.5523 or 650.833.7100 (outside the USA)

DevX.com MarketPlace prices are subject to change without notice. The Development Exchange, DevX, and DevX.com are trademarks of Fawcette Technical Publications. All products listed are registered marks or trademarks of their respective owners.

http://books.devx.com

theme. Coordinate storage rests on the idea that sparse matrix storage is a matter of storing the least amount of information needed to represent the matrix. On first glance, then, it seems that all we need to store for each nonzero element is its row coordinate, column coordinate, and data value. So, we could store our sparse matrix in three arrays: row[], col[], and val[]. The ith nonzero value would be stored by placing its row coordinate in row[i], its column coordinate in col[i], and its value in val[i].

What's important, of course, is how the nonzero items are arranged in the arrays. A purely random ordering of nonzero elements isn't worth much—searching for an item would require the code to walk through the array from one end to the other. (Since most searches into the arrays would turn up empty, a lot of time would be uselessly spent.)

Obviously, we should arrange the items either in row order or column order. Which arrangement we choose will be guided by the nature of the algorithms that access the matrix. Still, even if the elements were arranged in row order, locating the start of an arbitrary row would require a linear search into the row[] array.

It turns out that, with a minor modification, we can do the coordinate storage technique one better (both improving access and reducing memory needed). Suppose that you've got a sparse matrix that you want to store using a coordinate technique. Let's presume that it's a small sparse matrix:

Continued from page 62.

```
    while(next.right != elem)
      next = next.right;
    next.right = elem.right;
    // Special case
    if(next.row == -1)
      next.right = (sparseElement)null;
  }

  // Return a sparseElement at coordinates given. Go by row.
  // Return null if not in matrix
  public sparseElement atRow(int row, int col)
  {
    sparseElement next;

    // Verify in bounds. Return 0.0 if out of bounds
    if((row < 0 ) || (row > rowmax) ||
      (col < 0) || (col > colmax))
      return((sparseElement)null);

    next = rowBase[row].right;

    // Empty row?
    if(next == (sparseElement)null)
      return ((sparseElement)null);
    while((next.col < col) &&
          (next.col != -1))
      next = next.right;

    if(next.col == col)
      return next;
    else
      return((sparseElement)null);
  }

  // Return a sparseElement value at coordinates given.
  // Go by row.
  public double valRow(int row, int col)
  {
    sparseElement get;

    get=atRow(row,col);
    if(get==(sparseElement)null)
      return((double)0.0);

    return(get.val);
  }

  // Return a sparseElement at coordinates given.
  // Go by column. Return null if not in matrix
  public sparseElement atCol(int row, int col)
  {
    sparseElement next;

    // Verify in bounds
    if((row < 0 ) || (row > rowmax) ||
      (col < 0) || (col > colmax))
      return((sparseElement)null);

    next = colBase[col].down;

    // Empty col?
    if(next == (sparseElement)null)
      return((sparseElement)null);
    while((next.row < row) &&
          (next.row != -1))
      next = next.down;

    if(next.row == row)
      return next;
    else
      return((sparseElement)null);
  }

  // Return a sparseElement value at coordinates given.
  // Go by column.
  public double valCol(int row, int col)
  {
    sparseElement get;

    get=atCol(row,col);
    if(get==(sparseElement)null)
      return((double)0.0);

    return(get.val);
  }

  // Return the first sparseElement in a row.
  // Returns null if empty row
  public sparseElement firstInRow(int row)
  {
    return(rowBase[row].right);
  }

  // Return the next sparseElement in a row.
  // Returns null if at end of row
  public sparseElement
    nextInRow(sparseElement elem)
  {
    elem = elem.right;
    if(elem.row == -1)
      elem=(sparseElement)null;
    return(elem);
  }

  // Return the first sparseElement in a column.
  // Returns null if empty column
  public sparseElement firstInCol(int col)
  {
    return(colBase[col].down);
  }

  // Return the next sparseElement in a column.
  // Returns null if at end of column.
  public sparseElement
    nextInCol(sparseElement elem)
  {
    elem = elem.down;
    if(elem.col == -1)
      elem=(sparseElement)null;
    return(elem);
  }
}
```

SUBSCRIBE

To the **Ultimate Add-on Tool** for Microsoft **Visual Basic®**

Get the nuts and bolts programming information and advice you can utilize right away. More Windows® programmers turn to *VBPJ* each month for practical pointers, in-depth technical articles, and solid information. Get twelve information-packed issues with your paid subscription. Plus you'll also receive:

- **TWO BONUS ISSUES**
  The annual *VBPJ* Buyer's Guide and the Enterprise Computing issues.*
- **101+ TECH TIPS**
  Companion booklet mailed with our February and August issues of *VBPJ*.
- **PREMIER CLUB DISCOUNT**
  25% off to *VBPJ*, *WEB Builder*, and *Java™ Pro* subscribers.

*Buyer's Guide is mailed in April; the Enterprise Issue is mailed in September.

WHAT YOU CAN **EXPECT** EACH MONTH **FROM *VBPJ*:**

- Access data faster, better from your VB apps
- Use OOP to create more efficient, maintainable code
- Find and use the best database tools and libraries
- Build Web sites and Web apps with VB
- Integrate multiple databases and technologies
- Learn how to use Visual Studio tools in your Windows apps
- Write tight, fast code
- Get the most out of VBA and Office
- Manage your code more efficiently
- Optimize resource and memory use

For **FASTER** Service Call:

1-800-848-5523

(Monday through Friday 9:00am-5:00pm PST)

INTERNATIONAL: 650-833-7100

FTP FAWCETTE TECHNICAL PUBLICATIONS

www.devx.com

or email us at **customerservice@fawcette.com**

Please allow four weeks for delivery of first issue. International subscriptions must be payable in U.S.Dollars plus postage: Canada/Mexico add $18 per year for surface delivery; all other countries add $44 per year for airmail. Visual Basic, ActiveX and C++ are trademarks of Microsoft Corporation. Java is a trademark of Sun Microsystems.All other trademarks are the property of their respective owners.

```
0 0 1 0
2 0 0 3
0 3 3 0
2 0 0 0
```

Storing this matrix by row using coordinate storage would look like this:

```
row[] = { 0, 1, 1, 2, 2, 3 }
col[] = { 2, 0, 3, 1, 2, 0 }
val[] =  { 1, 2, 3, 3, 3, 2 }
```

Notice that the row[] array has repeated entries for rows 1 and 2. We can eliminate the extra entries if we use what's called (in this instance) compressed row storage. It's a simple idea, and it hinges on turning the row[] array into an array of (gasp) pointers into the col[] and val[] arrays. It requires a change only in the row[] array:

```
row[] = {0, 1, 3, 5 }
col[] = { 2, 0, 3, 1, 2, 0 }
val[] =  { 1, 2, 3, 3, 3, 2 }
```

Now, the ith entry in the row[] array corresponds to the ith row (rather than the ith nonzero value in the sparse matrix). In addition, each entry in row[] holds an index into the col[] and val[] arrays where the nonzero column coordinate and data values are stored.

Read it like this: row[]=0 means that col[0] holds the column coordinate and val[0] holds the value for the first nonzero element in the first row. row[1]=1 means that col[1] holds the column coordinate and val[1] holds the value for the first nonzero element in the second row. row[2]=3 means that the information (column coordinates and values) for row three "starts" in col[3] and val[3], respectively. (This also means that the second row has two nonzero values.)

In short, the row[] array has only as many entries as there are rows in the matrix—rather than as many as there are nonzero values. Although in our example the storage saving isn't much, it can amount to several hundred KB on, say, a 100,000×100,000 sparse matrix (depending on how sparse the matrix is).

Finally, locating the start of a particular row is fast. In the matrix multiplication example, if you stored the A matrix in compressed row fashion and the B matrix in compressed column fashion, the multiplication algorithm becomes simple and fast. In fact, this technique is used by the SciMark Java benchmark (found at http://math.nist.gov/scimark/).

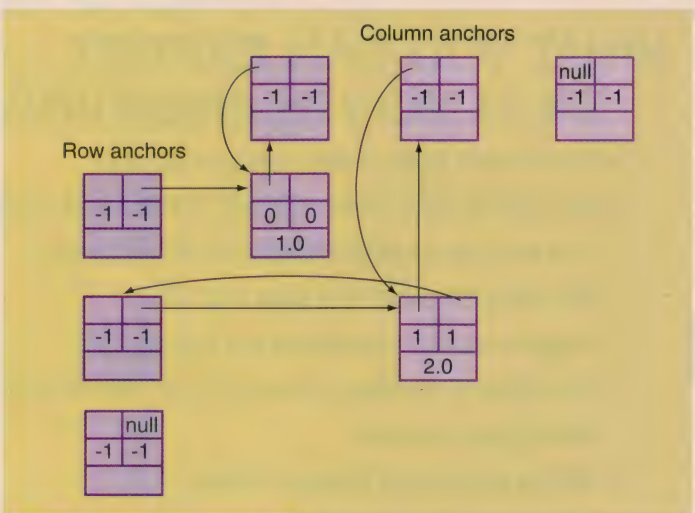


Figure 1 A 3×3 Array in Java This is the sparse matrix data structure implemented by the code in Listing 2. This structure is based on a structure presented by Donald Knuth (see text). Knuth's structure has the row and column anchors pointing to the last elements in the row and column; this structure points them to the first. (Knuth arranged his structure to simplify the coding of a pivot algorithm that appears in conjunction with the data structure.) This structure provides good sequential access of matrix elements: that is, you can quickly find the first and next elements in a row or column. Random access is not so fast, since it requires walking lists that can be arbitrarily long.

Lava Rocks

A final storage technique for sparse arrays that I want to mention is one I stumbled across on the Internet. This technique is probably more accurately called a virtual paged byte array; it was devised by Charlton Rose for his "Lava Rocks" Java class library (available at http://sharkysoft.com).

Lava Rocks includes useful classes, such as a Java equivalent to printf(), file utilities for reading and writing .WAV files, and classes for manipulating arrays. The Lava Rocks ByteArray32 class creates a large (2^{32} entry) virtual byte array.

When you create a ByteArray32 object, it builds a 256-element array of references to objects of class ByteArray24. An object of class ByteArray24 creates a 256-element array of objects of class ByteArray16, each object of which creates a 256-element array of objects of class ByteArray8—which finally creates an array of 256 bytes. The result of these nested arrays—if you can hold it all in your head—is a virtual array of $256\times256\times256\times256=2^{32}$ bytes.

However, what makes this a virtual array is that, upon creation, only the "topmost" array (ByteArray32) is actually instantiated—and the members of that array are null. No actual byte array is allocated. A read operation on the array returns 0. (The ByteArray32 get() method returns 0 for any byte element requested that is not yet allocated.)

The "lowmost" byte arrays that hold nonzero values—and the intervening reference arrays—are allocated only when a nonzero byte is written. The ByteArray32 set() method handles all the appropriate housekeeping of allocating (instantiating, if you will) the necessary arrays. The upshot is that you have a kind of "paged" virtual array (a page being a block of 256 bytes) of 2^{32} bytes with reasonably quick access speed, given that a search for an item requires following a chain of up to four references.

While the Lava Rocks' "virtual paged" array is specialized to bytes, it would be an easy matter to extend it to other data types. In addition, it's really a large 1D array of bytes, and its allocation is coarser than the other techniques I've shown (it allocates whole rows at a time).

Still, even if the Lava Rocks byte array doesn't solve your problem, there are enough techniques to choose from so that if you ever do encounter a sparse array, the odds are good that you can deal with it without exhausting memory. ■

Rick Grehan is a technical writer with Compuware/NuMega. Rick was a Senior Editor at *BYTE* magazine (where he wrote the JavaTalk column) and at *Computer Design* magazine. He can be reached at rgrehan@top.monad.net.

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0002 Download all of the code for this issue of *Java Pro*. Free online registration is required.
JP0002IT DevX Premier Club members download the code for this article.
JP0002IT_T DevX Premier Club members can read this feature online.

Not a member yet? Go to www.DevX.com to sign up for the free Registered Level or to subscribe to the Premier Club.

Build a better, smarter Web site

Only $29.95 for four quarterly issues—you'll save 42% off the newstand price!

With *WEB Builder CD*, your complete Web Development tool kit

All the Web development resources you need to take your site to the next level! Every issue delivers:

- Critical how-to articles from *VBPJ, Java™Pro, WEB Builder, Getting Started with Visual Basic,* and *Enterprise Development* magazines
- The latest versions of new tools and utilities
- Dozens of updated product reviews and fresh links

Plus, get valuable resources to continue and expand learning: links, software, tips, basics, code and more!

Subscribe Today—4 issues for $29.95

Order online at www.devx.com
Or call 1-800-848-5523 International orders: 650-833-7100
Fax 650-321-3818

FTP FAWCETTE TECHNICAL PUBLICATIONS 209 Hamilton Ave., Palo Alto, CA 94301

Visual Basic is a registered trademark of Microsoft Corporation. Java is a trademark of Sun Microsystems. *VBPJ* is a trademark of Fawcette Technical Publications.

Continued from page 72.

A At least one checkbox in a CheckboxGroup must be selected at all times. A CheckboxGroup mimics a set of radio buttons, where the pressing of a button results in the ejection of all other buttons. In general, if you require all of your checkboxes to be deselected, it is possible that they really should not be in a CheckboxGroup. However, if the options represented by the checkboxes are indeed mutually exclusive, and you really would like to be able to deselect all of them, you can use the following trick: Add an extra checkbox to the CheckboxGroup, but don't add it to a container component and don't display it. When you want to deselect all the visible checkboxes, just set the invisible checkbox's state to true. From a user interface design standpoint, it may be better to make this extra button visible and give it a label like "None" so that it is clear to the user that it is possible not to choose any of the other options.

Q When a request message is processed through the HttpServlet doGet() method, who sets the parameters and attributes for the HttpServletRequest argument? What is the difference between an attribute and a parameter?

A Client requests are passed on to a servlet through the Web server's servlet engine. The servlet engine packages the request into a ServletRequest object, through which request attributes and parameters can be extracted. The HttpServlet class examines the request and repackages it as an HttpServletRequest before dispatching it to one of its do() methods, such as doGet(). Attributes provide a means for the Web server to give a servlet extra information about a request. They do not represent data that was sent directly from the client. Parameters are akin to function arguments. They are name value pairs transmitted by the client and interpreted by the servlet. For example, the values submitted through an HTML form would be passed as parameters that could be accessed through the HttpServletRequest argument to doGet().

Q Is there a way to declare and invoke methods in JavaServer Pages?

A Methods can be declared in the same way as variables in JavaServer Pages. It can help to think of JSP as a convenient syntax for defining servlets. Any variables or methods that you declare are members of the servlet. The syntax for making declarations in JSP is: <%! declaration %> You can access the declared variables and methods in any expression or scriptlet block. See Listing 1 for an example.

Listing 1 I Do Declare This example shows how to declare a method and invoke it in a JSP expression.

```
<html>
<body>

<!-- Declare a method. -->

<%!
  public String aMethod() {
    return "<b>A String.</b>";
  }
%>

This JSP page demonstrates how to declare and invoke a
method.
The next line is the output of calling aMethod():
<br>
<p>
<!-- Inline the output of an expression. -->

<%= aMethod() %>

</body>
</html>
```

Q We use applets to allow users to access a database, but are looking for ways to improve performance. Is a servlet-based solution faster than an applet-based solution?

A This will depend on the application, the number of concurrent clients, the average network latency and bandwidth between a client and the server, and other factors. In general, if you want your application to scale to large numbers of users, you will want to forego the applet-centric solution. If each client applet has to create a separate connection to the database, transfer data, and process it locally, you face several potential bottlenecks. First, you could overload your database with too many connections. Second, you may have to transfer a lot of data over the network, which may take a long time. Third, the client may not have enough processing power or memory to process all of the data in a timely manner, if at all. Other problems also exist, such as the inability to centralize business logic, leading to software maintenance difficulties.

Always recompile code that may be dependent on a modified class

The preferred design for such a system is a three-tier architecture. A three-tier system consists of a client, a middle layer that performs transaction processing, and a back-end server. The client is usually lightweight, able only to issue queries and receive final results. An example of this would be a simple HTML form. The middle layer's job is to mediate access to server resources and possibly perform processing on the client's behalf. The server, of course, stores the database or other service that the client ultimately wants to access. Additional data processing can happen on the server and extra tiers can be added if multiple resources need to be accessed.

Multitier architectures offer several advantages. The primary advantage is the ability to scale to larger numbers of clients. The transaction processing layer makes this possible. It can keep a pool of open connections to the server, using them as appropriate to serve client requests, saving the overhead of opening a new connection for every client. Rather than overloading the server with queries, the transaction processor can throttle back the load on the server by serializing requests. Data processing can also be removed from the client and pushed closer to the data. The second major advantage is the ability to subdivide software development cleanly. Each layer performs a well-specified task that can be developed without overly impacting the other layers.

Q I recently heard Enterprise JavaBeans are not fully portable. Is this true?

A The Enterprise JavaBeans specification allows the EJB Container Provider to use either the JDK 1.1 or Java 2 APIs. This introduces the possibility that Enterprise JavaBeans won't run in all EJB runtime environments. To guarantee portability, you should restrict yourself to the 1.1 APIs. However, I would venture that most vendors will support Java 2, and it is probably not worth sacrificing the use of Java 2 features for the sake of guaranteed portability.

Other characteristics of the EJB runtime environment affect EJB portability. For example, an EJB Container may use multiple JVMs to execute Beans, which would render reliance on single JVM synchronization mechanisms and shared static variables nonportable. For a complete description of the programming guidelines you should follow to write portable Enterprise JavaBeans, read Chapter 18 of the Enterprise JavaBeans Specification.

Q How do you determine when a client has closed its socket so that a server can cease to wait for messages from that client?

A When a socket endpoint closes its socket, a read on the InputStream at the other end will return –1, indicating end of file. If the process crashes, the same will happen because the operating system will close the socket when the process dies. However, if the

Listing 2 The Reconstruction Era This code gives an example of how to convert a class instance to a byte array and then reconstruct the original class from the byte array.

```
import java.io.*;

public class ByteSerialize {

  public static void main(String[] args) {
    Integer write, read;
    ObjectOutputStream out;
    ObjectInputStream in;
    ByteArrayOutputStream byteStream;
    byte[] bytes;

    write = new Integer(5);

    System.out.println("write: " + write);

    try {
      byteStream = new ByteArrayOutputStream();
      out = new ObjectOutputStream(byteStream);
      out.writeObject(write);
      out.close();

      // Byte representation of integer
      bytes = byteStream.toByteArray();

      in = new ObjectInputStream(new ByteArrayInputStream(bytes));
      read = (Integer)in.readObject();
      in.close();
    } catch(IOException ioe) {
      ioe.printStackTrace();
      return;
    } catch(ClassNotFoundException cnfe) {
      cnfe.printStackTrace();
      return;
    } catch(ClassCastException cce) {
      cce.printStackTrace();
      return;
    }

    System.out.println("read : " + read);

  }
}
```


Product Showcase

JAVAPro

Linux • OS/2 • Solaris • NT
HP-UX • MacOS • AIX
Simplicity for Java
Build professional Java apps visually on the platform of your choice.
FREE TRYOUT - 100% Pure Java IDE
http://www.datarepresentations.com

INTEROPERABILITY.
Edge Server™
Imagine streaming to Java™ clients, QuickTime™, RealPlayer™, and MS Media Player™ from one appliance. GMV Network EdgeServer™ supports Java™ Media Framework and is capable of supporting multiple player platforms.
Welcome to true streaming media interoperability.
GMV NETWORK
18 S. Michigan Ave. Suite 814
Chicago, IL 60603
312.357.0411
gmvnetwork.com

NEW VERSION 4.0 - FREE TRIAL AT WWW.DEV-SOFT.COM
IP*Works!™
20 PURE JAVA BEANS FOR THE INTERNET
SMTP NNTP FTP IPPORT
POP SNMP HTTP IPDAEMON
IMAP NETCODE SNPP UDPPORT
LDAP MIME MX MULTICAST
REXEC RSHELL TELNET IPINFO
ALSO AVAILABLE AS ACTIVEX CONTROLS, C++ CLASSES, C LIBRARIES, AND NATIVE DELPHI AND C++ BUILDER VCLs
www.dev-soft.com

Java Jam 2000
A Seven-Night Caribbean Journey November 2000
geekcruises.com

SPELL CHECK!
Add a powerful, professional quality spell checker to Java applications and applets. Visit our Web site for information and demos.
Wintertree Software Inc.
1-800-340-8803
613-825-6271
www.wintertreesoftware.com

Listing 3 Testing the Behavior This program enables you to test the behavior of java.awt.Scrollbar compared to javax.swing.JScrollBar.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class ScrollTest {

  public static class ScrollTestComponent extends JPanel
    implements AdjustmentListener
  {
    JTextArea _scrollText;
    Scrollbar _scrollBar;
    JScrollBar _jscrollBar;

    public ScrollTestComponent() {
      _scrollBar = new Scrollbar();
      _jscrollBar = new JScrollBar();
      _scrollBar.addAdjustmentListener(this);
      _jscrollBar.addAdjustmentListener(this);
      _scrollText = new JTextArea(40, 40);
      _scrollText.setEditable(false);
      setLayout(new BorderLayout());
      add(_scrollBar, BorderLayout.WEST);
      add(_scrollText, BorderLayout.CENTER);
      add(_jscrollBar, BorderLayout.EAST);
      printScrollValues();
    }

    public void printScrollValues() {
      _scrollText.setText("    ScrollBar: ");

_scrollText.append(Integer.toString(_scrollBar.getValue()));
      _scrollText.append("\n    JScrollBar: ");

_scrollText.append(Integer.toString(_jscrollBar.getValue()));
    }

    public void adjustmentValueChanged(AdjustmentEvent e) {
      printScrollValues();
    }

  }

  public static void main(String[] args) {
    JFrame frame = new JFrame("Scroll Test");
    WindowListener exitListener;

    exitListener = new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
        Window window = e.getWindow();
        window.setVisible(false);
        window.dispose();
        System.exit(0);
      }
    };

    frame.addWindowListener(exitListener);

    frame.getContentPane().add(new ScrollTestComponent(),
      BorderLayout.CENTER);
    frame.pack();
    frame.setVisible(true);
  }
}
```

process's host crashes, there is no way to detect the endpoint termination. With native system programming, it is possible to set the SO_KEEPALIVE socket option to test the integrity of a connection on a more frequent basis than the default two hours. Java does not allow you to do this, and instead relies on the system default.

Q Can I finalize an object without waiting for the garbage collector?

A The System class contains a convenience method called gc() that can be used to ask the JVM to run the garbage collector. It also contains a method called runFinalization(), which can be invoked to ask the JVM to perform all pending finalizations. These methods are the equivalent of calling Runtime.getRuntime() and invoking the corresponding method on the resulting Runtime instance. To finalize objects that are no longer referenced, you would call System.gc() to find all unreferenced objects and then System.runFinalization() to perform finalizations. However, garbage collection and finalization are not guaranteed to occur after calling these methods. The methods only specify that the JVM will make a best effort to complete the requested actions.

Q How can I convert a class instance to a byte array and then subsequently reconstruct the original class from the byte array?

A Java's Object Serialization API allows you to write class instances to streams. To store a class instance in a byte array, you can wrap a ByteArrayOutputStream with an ObjectOutputStream. To read a class instance from a byte array, you can wrap an ObjectInputStream with a ByteArrayInputStream. You just have to remember that all serializable objects must implement the Serializable interface. Listing 2 demonstrates how to do this.

Q I have an application that uses dialup networking. If I connect after starting my application, InetAddress.getLocalHost() always returns 127.0.0.1. Even if I connect before starting up, then disconnect and reconnect, the IP address I was assigned on the first connection by PPP is returned and not the new one. Why does this happen?

A I have taken some jabs at the java.net package, either from a design perspective or to criticize the JDK implementation. You've just run into another poorly conceived implementation detail. The InetAddress class in JDKs 1.0.2 and 1.1.x misguidedly caches hostname lookups in a hashtable. When a lookup is performed, the cache is checked first. If the entry is not in the cache, native DNS resolution is performed.

The flaw with this implementation is that it leads exactly to the problem you've encountered. If the hostname to IP address mapping of the local host changes, it will not be reported. In principle, it may be all right to do this because DNS servers cache mappings for set periods of time and will not reflect changes instantly. But long-lived Java applications will become victims of outdated information stored in their hashtable. JDK 1.2 tries to improve the situation by storing cache entries in a HashMap and giving each cache entry an expiration time. However, the fundamental problem is still there.

Rather than provide an extra layer of name resolution caching in InetAddress, the JDK implementation should rely on the native name resolution mechanisms that already have performance optimizations built in. In my opinion, you're dealing with a bug and the only way around it is to fiddle with the JDK classes. The JDK 1.2 implementation should at least give you control over the caching policy and expiration time, but it doesn't.

Q Being an advanced C++ user, I like to simplify code and often use operator overloading. Is it possible to overload operators in Java?

A C++ programmers moving to Java ask this question frequently, so it bears repeating the answer. Java does not support operator overloading. Operator overloading is mostly a syntactic convenience and could be added to the language relatively easily. The Java Community Process has no proposals to add op-

erator overloading to the language, but a generic types proposal has been approved, so it is possible that operator overloading could make its way into the language. For the moment, you either have to live without operator overloading, or try out a nonstandard Java derivative, such as the JFront preprocessor (www.winternet.com/~gginc/jfront/), which adds limited support for operator overloading.

Q Can I dynamically determine the value of a Scrollbar when I scroll? The adjustmentValueChanged() method gets called only when you stop scrolling. But I want to catch the value of the scrollbar when I press on the scrollbar and drag it. Is there a way to do this?

A You pay a price for the cross-platform portability of Java. Your code may compile once and run everywhere, but it may not behave the same way on every platform. The Scrollbar class is an AWT component whose ultimate representation and behavior is determined by a native peer. In Windows 95, you may find that the behavior of a Scrollbar is to notify its listeners of a change in value only after it has finished moving, rather than smoothly as it is moving. In a Unix/Motif environment, I find that each time the Scrollbar moves a unit increment, the listener is notified. Rather than rely on native peers, it is better to use the Java Foundation Classes, specifically the Swing package, to guarantee consistent behavior across platforms. The example program shown in Listing 3 allows you to test the behavior of java.awt.Scrollbar versus that of javax.swing.JScrollBar. A listener is registered with a scrollbar of each type and prints out the scrollbar values each time it is notified of a change through the AdjustmentListener interface.

Q Does each instance of a class have its own copies of instance methods in the same way it has its own copies of instance variables?

A Every time you instantiate a class, new memory is allocated to store the object. Part of this storage includes space for copies of its member variables. Rather than create a new copy of each method into a class instance, which would be expensive, a virtual lookup table mapping methods to code blocks is allocated. You can think of class methods as functions that take an invisible argument that references the invoking instance—that is, the "this" reference. That makes it possible to store methods independent of instantiated classes. When a class instance invokes a method, a lookup is performed in its virtual function table to find the proper code block to execute. Therefore, all class instances share the same code, but have their own data. Novel runtime optimization techniques have been applied to improve the performance of virtual function lookups in Java so that you only pay a lookup penalty the first time you invoke a method. ■

Daniel F. Savarese is a partner in QuaTeams, LLC, a consulting firm specializing in advanced information technology services. He has worked as a senior scientist at the California Institute of Technology and Director of Java Product Development at ORO, Inc. Daniel has written several Java packages, including the OROMatcher and PerlTools regular expression libraries and the NetComponents network protocol library. He is also a co-author of "How to Build a Beowulf." Daniel can be reached at dfs@savarese.org.

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0002 Download all of the code for this issue of *Java Pro*. Free online registration is required.
JP0002AP DevX Premier Club members download the code for this article.
JP0002AP_T DevX Premier Club members can read this feature online.

Not a member yet? Go to www.DevX.com to sign up for the free Registered Level or to subscribe to the Premier Club.

Advertiser Index

Ad Astra www.jumpingbeans.com	17	**Exchange & Outlook Magazine Subscriptions** 800-848-5523 www.exchange.devx.com	59	**RSA Security** 800-782-5453 www.rsa.com/win	21	**Showcase**	
Apachecon 2000 www.apachecon.com	55	**InformIT** www.informit.com	31	**SilverStream Software Inc.** www.silverstream.com	10	**Data Representations** 888-307-9550 www.datarepresentations.com	69
Applied Reasoning 800-260-2772 www.appliedreasoning.com	8	**IONA Technologies** 800-672-4948 www.IONA.com	6, 7	**Sybase** 800-8-SYBASE (878-2273) www.sybase.com	37	**DevSoft** 919-493-5805 www.dev-soft.com	69
BEA Systems 800-WEBLOGIC(932-5644) weblogic.beasys.com	13	**inquiry.com** 800-848-5523 www.inquiry.com	51	**Tidestone Technologies** 800-884-8665 www.tidestone.com	2, 3	**Geek Cruises** www.geekcruises.com	69
Career Central www.careercentral.com/database	25	**Java Pro Magazine Subscriptions** 800-848-5523 www.java-pro.com	24	**TogetherSoft** www.togethersoft.com	C2	**GMV Network LLC** www.gmvnetwork.com	69
Computerjobs.com www.computerjobs.com	43	**KL Group** 800-663-4723 www.klgroup.com	1	**Visual C++ Developer's Journal Magazine Subscriptions** 800-848-5523 www.vcdj.com	49	**Wintertree Software Inc.** 800-340-8803 www.wintertree-software.com	69
Compuware Corporation 800-4-NUMEGA (468-6342) www.compuware.com/numega	23	**LOOX Software** 800-684-5669 www.loox.com	33	**Visual Basic Programmer's Journal Magazine Subscriptions** 800-848-5523 www.vbpj.com	65		
DEVELOPMENTOR 800-699-1932 www.develop.com	27	**ParaSoft Corporation** 888-305-0041 www.parasoft.com/superior.htm	45	**YOUcentric** 888-462-6748 www.youcentric.com/ecrm	15		
DevX Books 800-848-5523 marketplace.devx.com	44, 63	**PreEmptive Solutions, Inc.** 800-996-4556 www.preemptive.com	C3	**Web Builder CD** 800-848-5523 www.devx.com	67		
DevX E-mail Newsletters 800-848-5523 www.devx.com	38	**Premier Club** 800-848-5523 www.premier-club.com	57	**Zero G** 415-512-7771 www.ZeroG.com	4		
DevX Java-Zone 800-848-5523 www.java-zone.com	53	**ProtoView Development Corp.** 800-231-8588 www.protoview.com	C4				
Enterprise Development Magazine Subscriptions 800-848-5523 www.enterprisedev.com	16						

Product Information

- To get immediate details on products advertised in this issue, you can simply log onto our online services at **http://infoasap.devx.com**.
- This provides a list of direct links to this month's advertiser's Web sites.

Advertising Sales Contacts

- **Kelly Keane, Western Regional Sales Manager**
  650-833-7128; kkeane@fawcette.com
- **Shana Deane, Eastern Regional Sales Manager**
  413-665-6910; sdeane@fawcette.com

This listing is provided as a courtesy to our readers and advertisers. The publisher assumes no responsibility for errors or omissions.
Sales Department: 209 Hamilton Avenue • Palo Alto, CA, USA 94301-2500 • 650-833-7100

Ask the Java Pro

Inveighing against Insidious Inlining

Despite productivity gains, Java programming is not without its share of headaches. I get bitten by unexpected problems every time a new JDK is released. My latest JDK upgrade trouble is the sort of thing that makes me think that someone has it in for me. Usually I can take comfort in the fact that at least a handful, if not a horde, of programmers around the world has encountered a problem before me. Sometimes, though, I suspect a conspiracy of silence over some particularly painful bugs.

The detailed workings of the Java compiler remain a mystery. Whereas C and C++ compilers possess many optimization options, the JDK compiler has only one, the –O flag. What exactly the –O flag does is not well documented. Among its many effects is the ability to inline functions under certain conditions. Typically, if the source code for a class is available to the JDK compiler while it compiles another class, it will attempt to inline invocations of final methods. Given Java's dynamic binding, this behavior is suspect. Suppose you change the method and recompile its class, but don't recompile the calling class. The calling class will now execute incorrectly, a situation that could have been avoided by not inlining the method. The lesson here is that you should not rely on dynamic binding to allow you to upgrade a library transparently without recompiling an application. Always recompile code that may be dependent on a modified class.

Cross-platform testing is just as important in Java as in any other language

BY DANIEL F. SAVARESE

An equally vexing problem is a direct byproduct of the method inlining performed by the JDK compiler's –O option and an unexpected interaction with the Java Security Manager. As you likely know, one of Java's more highly touted features is its capability for security. When a class is loaded or code is executed, all sorts of security checks can occur to ensure that no untrusted actions occur.

Part of the security checking process is class verification. Class verification ensures that a class file is properly formatted and does not perform any operations disallowed by the Java virtual machine. Unlike C++, where language-specified protections such as class member access are enforced at compile time, Java verifies some of these things at runtime. For example, a top-level class is not allowed to access another top-level class's private members. Even though a Java compiler is not supposed to generate such code, it is possible to construct a class file where this occurs. When loading a class for the first time, the JVM class verifier can check to see if any access violations occur within a class, and, if so, abort execution.

The key word in that last sentence is *can*. The JVM class verifier *can* check for access violations, but may not necessarily do so. As it turns out, the JDK 1.1.x runtimes did not check for access violations, or at least for classes illegally accessing the private member variables of other classes. I discovered this oversight when I started getting a stream of bug reports from developers using code that I had developed with the JDK 1.1.7 compiler. They all complained that the JDK 1.2 runtime (still in beta at the time) threw an IllegalAccessError when executing my code.

It turned out that the JDK 1.1.7 compiler was overzealous in its optimizations, generating class files that wouldn't pass the JDK 1.2 runtime's class verifier. The culprit was a final method that returned the value of a private variable. The 1.1.7 compiler inlined this method when called from another class, creating an illegal access to a private member variable in the class file's byte code. The 1.1.7 runtime never detected this because its class verifier did not check for this particular access violation. However, the 1.2 runtime did perform this check, causing a lot of problems for developers and making me look bad in the process.

I could have detected the problem if I had upgraded my development environment more aggressively at the time or at least performed regression testing each time a new JDK was released. I had not thought to do this because we're led to believe that all of our Java code is write once, run anywhere. But that's clearly not always the case. We are also led to believe that Java enhances our productivity, not only because of the language design, but also because of its cross-platform nature. However, cross-platform testing is just as important in Java as in any other language. Testing across multiple JVMs on the same platform is also incredibly important. All that development time you save may just balance out the added time the software testing division has to spend to make sure your code works everywhere.

Q How can I put all the checkboxes in a CheckboxGroup into an unselected state when Checkbox.setState(false) does not automatically deselect the checkbox once it has been selected.

Continued on page 68.

Your customers don't like to wait!

(neither do bosses)

DashO™ PRO

REDUCE JAVA APPLET/APPLICATION SIZE UP TO 70%!

3 PRODUCTS IN 1!

DECREASE DOWNLOAD TIMES
DashO-Pro analyzes your program and extracts only the classes your application needs! It then removes unused methods and fields for unmatched space savings!

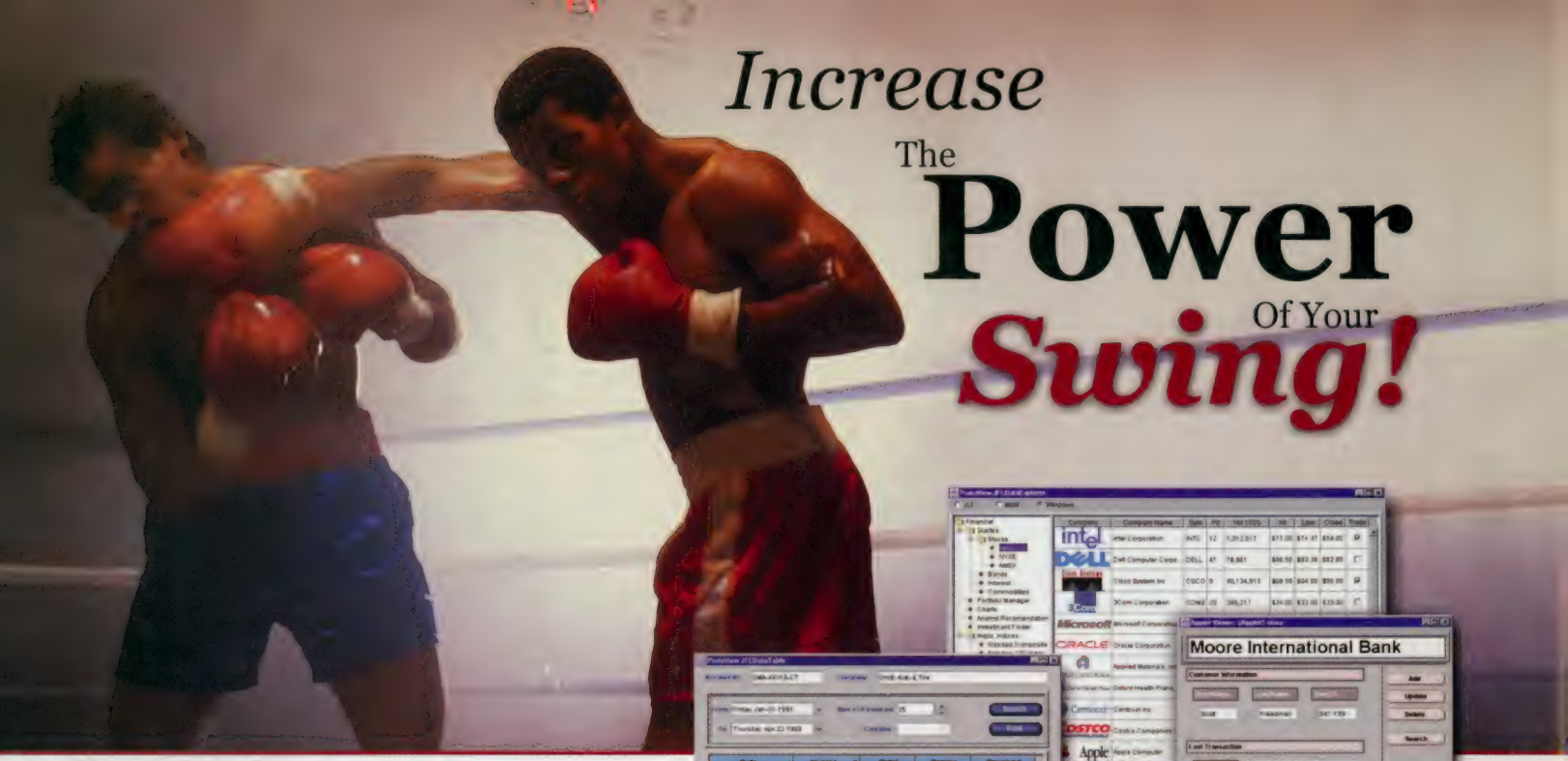
OBSFUCATE
DashO-Pro includes the patent pending Overload-Induction™ renaming system. No renamer saves more space or protects your code better!

OPTIMIZE
DashO-Pro actually optimizes your code for top performance!
Up to 30% faster!

preEmptive solutions

www.preemptive.com
1-800-996-4556

©1999 preEmptive Solutions, Incorporated. All Rights Reserved. Java is a trademark of Sun Microsystems. All trademarks are owned by their respective companies.



Increase The Power Of Your Swing!

Ladies & Gentlemen...

The First Drop-In Extensions For JFC

New JFCDataTable & JFCDataTree Extend JTable & JTree

The newest addition to the JFCSuite™, the JFCDataTable is the only true drop-in replacement for JTable. In minutes gain access to a wealth of features while maintaining all of your existing code.

Snap JFCDataTable Into Your Existing JTable Applications And Get:

- Easy-to-use printing.
- Rapid multi-column sorting.
- Advanced cell editing & formatting.
- Extended UI properties.
- Improved keyboard handling.
- Maintain existing JTable code.

Drop The New JFCDataTree Into Your Existing JTree Applications And Get:

- Extended node images.
- Improved keyboard handling.
- Customize color, font and border styles.
- Event listeners for drag & drop events.

Lightweight GUI Components With Heavyweight Features

The JFCSuite™ is a collection of professionally polished beans with advanced UI styles and easy-to-use data models.

Also Includes:

- ***JFCDataExplorer:*** winner of both the JDJ Editor's Choice and Java Pro Reader's Choice awards, this cutting-edge component simplifies the presentation of data using the classic Explorer-style user interface.
- ***JFCDataCalendar:*** full support for images in days, drop-down calendars, long date and International formats and multiple selection.
- ***JFCDataInput:*** over 12 components rich in data validation with a wide variety of professional quality GUI styles to choose from. Includes Mask Edit, Currency, Time, Numeric, Date and many more.

Get the JFCSuite today, and see why they're the hardest-hitting components on the market.

Judge For Yourself!
Download

Full-Featured Trial Versions!

Order Online!

or call: 800-231-8588

PROTOVIEW™
Software Tools for Component Based Development
www.protoview.com

Copyright © 1999 ProtoView Development Corporation All rights reserved. All ProtoView product names are trademarks of ProtoView Development Corporation. Sun, Sun Microsystems, Java, JavaBeans and Java Foundation Classes are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries., Sun Microsystems, Java, JavaBeans and Java Foundation Classes are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

JAVA Pro

YES ☑ Send me one year (13 issues) of *Java™ Pro* for only $29.95

Name _____

Address _____

City _____ State _____

Zip _____ Country _____

E-mail address _____

❑ Bill me ❑ Payment enclosed Charge my: ❑ Visa ❑ Master Card ❑ AMEX

Card # _____

Expiration date _____

Signature _____

One year is 12 issues plus one special issue. Savings based on annual cover price of $77.35. Please allow 4-6 weeks for delivery of your first issue. International subscriptions must be payable in US dollars plus postage: Canada/Mexico add $18/year for surface delivery; all other countries add $44/year for airmail delivery. Canadian GST included. 4114

SAVE 61% SAVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 Palo Alto, CA

POSTAGE WILL BE PAID BY ADDRESSEE

P.O. Box 54584
Boulder, CO 80323-4584

JAVA™Pro Savings Voucher

Get Java™ Pro for only $2.30 per issue—and save over 61% off the cover price. A single tip in one issue can more than pay for a year's subscription. Order Now and Save!....

Annual Cover Price	Your Price	Subscription Savings
$77.35	$29.95	61%

❑ **1 Year** (13 issues) $29.95—SAVE 61%

NAME_____

ADDRESS_____

CITY_____ STATE_____ ZIP_____

E-MAIL ADDRESS_____

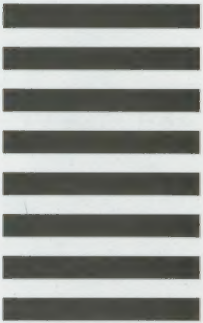
❑ Payment enclosed ❑ Bill me later

My Guarantee:
If at any time I do not agree that *Java™ Pro* is the best, most useful source of information on Java Programming, I can cancel my subscription and recieve a full refund.

One year is 12 monthly issues and one special issue. Savings based on annual cover price of $77.35. Please allow 4-6 weeks for delivery of first issue. International subscriptions must be payable in U.S. dollars plus postage. Canada and Mexico add $18/year, all other countries add $44/year for airmail delivery. Canadian GST included.

4113

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 Palo Alto, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVA™Pro

P.O. Box 54584
Boulder, CO 80323-4584